



CONTRIBUTIONS À L'ORDONNANCEMENT RÉACTIF DES INSTALLATIONS DE TRAITEMENT DE SURFACE – APPLICATION INDUSTRIELLE –

Etienne Chové

► To cite this version:

Etienne Chové. CONTRIBUTIONS À L'ORDONNANCEMENT RÉACTIF DES INSTALLATIONS DE TRAITEMENT DE SURFACE – APPLICATION INDUSTRIELLE –. Automatique / Robotique. Université de Nantes, 2010. Français. <tel-00533299>

HAL Id: tel-00533299

<https://tel.archives-ouvertes.fr/tel-00533299>

Submitted on 5 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE NANTES
ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES
DE L'INFORMATION ET DES MATERIAUX
Année : 2010

Thèse de Doctorat de l'Université de Nantes
Spécialité : Automatique et Informatique Appliquée

Présentée et soutenue publiquement par

Etienne CHOVÉ

le 20 octobre 2010

à l'Institut Universitaire de Technologie de Nantes

CONTRIBUTIONS A L'ORDONNANCEMENT REACTIF
DES INSTALLATIONS DE TRAITEMENT DE SURFACE
— APPLICATION INDUSTRIELLE —

JURY

Président :

Rapporteurs : Sylvie Norre, Professeur à l'Université Blaise Pascal, Clermont-Ferrand II
Damien Trentesaux, Professeur à l'Université de Valenciennes et de Hainaut-Cambrésis

Examineurs : Pascal Berruet, Professeur à l'Université de Bretagne Sud
Pierre Castagna, Professeur à l'Université de Nantes
Jean-Jacques Lesage, Professeur à l'École Normale Supérieure de Cachan
Sébastien Mary, Ingénieur Aerolia, Saint-Nazaire
André Thomas, Professeur à l'École Nationale Supérieure des Technologies et Industries du Bois, Épinal

Directeur de thèse : Pierre Castagna

Laboratoire : Institut de recherche en communications et cybernétique de Nantes

Composante de rattachement du directeur de thèse : Université de Nantes

N° ED 0366-xxx

*À Bérengère,
À Pauline.*

Remerciements

Je tiens en tout premier lieu à remercier *Pierre Castagna*, professeur à l'Université de Nantes et directeur de cette thèse, pour le temps qu'il a consacré à ces travaux. Une partie du chapitre 2 sur le pilotage réactif des lignes de traitement de surface est issue de ses travaux de recherche antérieurs. C'est grâce à ses contacts industriels et à sa disponibilité que les travaux de cette thèse ont pu parvenir à ces résultats.

Je remercie aussi vivement les ingénieurs d'Aerolia Saint-Nazaire (anciennement Airbus), notamment *Sébastien Mary*, pour la problématique passionnante qu'ils nous ont soumise. Qu'ils trouvent dans le nouveau système de pilotage de l'unité de traitement de surface, la récompense de leur confiance, de leur participation active et des heures de réunion, parfois nocturnes.

Je remercie tous les membres du jury d'avoir accepté de porter leur intérêt sur ces travaux et d'être venus, parfois de loin, assister à la soutenance. Je remercie tout spécialement *Sylvie Norre*, professeur à l'Université Blaise Pascal (Clermont Ferrand II), et *Damien Trentesaux*, professeur à l'Université de Valenciennes et de Hainaut-Cambrésis, d'avoir rapporté cette thèse.

Mes remerciements vont ensuite à tous les membres de l'équipe ACSED¹ de l'IRCCyN², dirigée par *Jean-Jaques Loiseau*, directeur de recherche CNRS. L'intérêt de tous pour les travaux de chacun développe une dynamique d'équipe qui a permis d'alimenter certaines idées développées dans cette thèse.

Cette thèse ne se serait aussi bien passée sans l'accueil chaleureux du département QLIO³ de l'IUT⁴ de Nantes, et notamment de son chef *Jean-Paul Guérin*, toujours attentif aux chercheurs qui occupent les locaux. Que ma petite participation à la vie du département leur laisse un bon souvenir de mon passage.

Un grand merci à *Christophe Plot*, ingénieur au département QLIO, pour les heures passées ensemble qui m'ont permis de me former dans beaucoup de domaines connexes à ces travaux (mécanique, informatique industrielle, électronique...).

Je remercie également tous les chercheurs rencontrés au fil des congrès, pour l'intérêt qu'ils ont porté à ces recherches. Je les remercie pour leurs remarques pertinentes qui ont permis l'avancée de ces travaux.

Je ne saurais oublier mon épouse *Bérengère* que je remercie pour son soutien et pour les nombreuses corrections qu'elle a apportées à ce manuscrit. Qu'elle reçoive, avec notre fille *Pauline*, mes vifs remerciements.

Enfin, je remercie mes parents, qu'ils voient dans l'aboutissement de mes études et de cette thèse, la réussite de leur mission.

¹ Analyse et Commande des Systèmes à Événements Discrets

² Institut de Recherche en Communications et Cybernétique de Nantes

³ Qualité, Logistique Industrielle et Organisation

⁴ Institut Universitaire de Technologie

Sommaire

| | |
|---|-----------|
| Remerciements..... | 1 |
| Introduction | 7 |
| Chapitre 1 : Évolution du pilotage des unités de production | 9 |
| 1. Évolution générale des systèmes organisés | 10 |
| 1.1. D'une organisation hiérarchique... .. | 10 |
| 1.2. ... à une organisation répartie | 11 |
| 1.2.1. Organisation hétérarchique | 12 |
| 1.2.2. Organisation holonique | 12 |
| 1.2.3. Organisation isoarchique..... | 13 |
| 1.3. Synthèse..... | 14 |
| 2. Évolution des systèmes de production | 14 |
| 3. Évolution des systèmes de pilotage..... | 17 |
| 3.1. Pilotage prédictif..... | 19 |
| 3.2. Pilotage dynamique | 20 |
| 3.3. Pilotage réactif | 23 |
| 3.4. Comparaison des méthodes de pilotage | 25 |
| Chapitre 2 : Ordonnancement des installations de traitement de surface | 27 |
| 1. Description des unités de traitement de surface..... | 28 |
| 1.1. Les porteurs | 28 |
| 1.2. Les cuves | 29 |
| 1.3. Les transporteurs (ponts roulants) | 31 |
| 1.4. Les navettes | 34 |
| 2. Problématique du Hoist Scheduling Problem..... | 35 |
| 2.1. Définition d'une gamme de traitement..... | 35 |
| 2.2. Notations..... | 36 |
| 2.2.1. Données du problème | 37 |
| 2.2.2. Résultat de l'ordonnancement..... | 37 |
| 2.3. Contraintes d'ordonnancement | 38 |
| 2.4. Fonction objectif | 39 |
| 2.5. Intégration du HSP dans les problèmes d'ordonnancement..... | 40 |

| | |
|--|------------------|
| 3. Classification des problèmes | 42 |
| 4. Approches de résolution | 45 |
| 4.1. Méthodes hors ligne | 45 |
| 4.1.1. Ordonnancement cyclique | 45 |
| 4.1.2. Ordonnancement prédictif | 47 |
| 4.2. Méthodes en ligne | 50 |
| 4.2.1. Ordonnancement dynamique (pilotage avec prévision) | 51 |
| 4.2.2. Ordonnancement réactif (pilotage sans prévision) | 52 |
| 4.3. Bilan sur les méthodes existantes | 53 |
| 5. Conclusion | 53 |
| <u>Chapitre 3 : Couplage entre pilotage réactif et pilotage prédictif</u> | <u>55</u> |
| 1. Présentation du problème | 56 |
| 1.1. Topologies étudiées | 56 |
| 1.2. Méthodologie pour la définition du pilotage réactif | 57 |
| 1.3. Présentation des cas industriels | 58 |
| 2. Proposition d'un pilotage réactif | 60 |
| 2.1. Principe du pilotage | 60 |
| 2.2. Définition des règles | 61 |
| 2.2.1. Règles produit/emplacement | 61 |
| 2.2.2. Règles produit/pont | 62 |
| 2.3. Modèle de simulation | 65 |
| 2.3.1. Objets physiques | 65 |
| 2.3.2. Objets virtuels | 68 |
| 2.3.3. Réalisation de la bibliothèque et exemple de modèle | 68 |
| 2.4. Exploitation des simulations | 71 |
| 2.4.1. Analyse graphique des résultats | 71 |
| 2.4.2. Indicateurs de performance | 72 |
| 2.5. Analyse des résultats | 74 |
| 3. Couplage entre pilotage réactif et prédictif | 76 |
| 3.1. Bref état de l'art | 76 |
| 3.2. Principe du couplage | 76 |
| 3.3. Modèle holonique | 78 |
| 3.4. Application à un cas industriel | 79 |

| | |
|--|------------------|
| 4. Implémentation..... | 80 |
| 4.1. Choix d'un outil d'ordonnancement..... | 80 |
| 4.2. Pilotage réactif ou pilotage dynamique ? | 81 |
| 4.3. Solution adoptée..... | 83 |
| 5. Conclusion | 83 |
| <u>Chapitre 4 : Optimisation de la topologie d'une unité de traitement de surface</u> | <u>85</u> |
| 1. La problématique | 86 |
| 1.1. L'implantation dans le cycle de vie de l'atelier | 86 |
| 1.2. Le champ des solutions d'implantation | 86 |
| 2. Quelques méthodes d'implantation | 88 |
| 3. Méthodologie de résolution : algorithme génétique..... | 89 |
| 3.1. Création d'une population initiale | 90 |
| 3.2. Mécanisme de reproduction/sélection | 91 |
| 3.3. Mécanisme de croisement..... | 92 |
| 3.4. Mécanisme de mutation | 93 |
| 3.5. Mécanisme de réparation..... | 94 |
| 3.6. Test d'arrêt..... | 94 |
| 4. Application à un cas industriel | 94 |
| 4.1. Présentation du cas industriel | 94 |
| 4.2. Exécution de l'algorithme | 95 |
| 4.3. Résultats..... | 96 |
| 5. Conclusion | 98 |
| <u>Chapitre 5 : Aide au lancement des produits dans un RHSP</u> | <u>99</u> |
| 1. La prise de décision dans un système réactif | 100 |
| 1.1. Exemple avec 2 produits | 100 |
| 1.2. Exemple avec n produits..... | 101 |
| 1.2.1. Définition des exemples | 101 |
| 1.2.2. Montage expérimental..... | 102 |
| 1.2.3. Résultats | 105 |
| 2. Heuristique d'exploration | 106 |
| 3. Résultats | 108 |
| 3.1. Résultat de l'heuristique | 108 |

| | |
|--|-------------------|
| 3.2. Étude du nombre de pas d'optimisation | 110 |
| 3.3. Étude du lancement successif de plusieurs instances de l'heuristique..... | 111 |
| 4. Conclusion..... | 113 |
| 5. Annexes : résultats du problème 2 | 114 |
| <u>Conclusion et perspectives.....</u> | <u>117</u> |
| <u>Liste des figures</u> | <u>119</u> |
| <u>Liste des tableaux</u> | <u>122</u> |
| <u>Bibliographie</u> | <u>123</u> |

Introduction

Dans un contexte industriel largement ouvert à l'étranger, les pays à faible coût de production imposent aux pays industrialisés une concurrence exigeante. Pour survivre et être compétitives, les entreprises doivent optimiser le trio *qualité-coût-délai*, cherchant ainsi à produire mieux, moins cher et/ou plus vite que leurs homologues. C'est dans ce contexte économique difficile que le développement de nouveaux procédés de fabrication et de nouvelles méthodes de pilotage permet de placer l'entreprise en haut du marché concurrentiel.

Dans ce contexte en perpétuelle évolution, l'horizon temporel se réduit. Là où Henry Ford avait imaginé un modèle unique de voiture pour permettre la production à la chaîne en grande série, faisant des prévisions sur plusieurs années, les entreprises actuelles doivent produire en parallèle des dizaines de modèles différents, à la demande du client. Dans cette situation, les prévisions sont très difficiles, et l'horizon de planification dépasse rarement le mois. Ainsi, dans certains cas, il n'est pas possible de préparer l'avenir. C'est alors que les nouvelles techniques d'ordonnancement sont nées. Les prévisions sont remplacées par la réaction du système à son environnement extérieur en mutation. Cette situation, difficile pour les entreprises, fait la joie de la recherche, permettant aux chercheurs de bâtir de nouvelles stratégies de pilotage, d'organisation, de méthodes de conception...

Nous exposerons dans le chapitre 1 de cette thèse les différentes évolutions des systèmes organisés, en observant les coïncidences et oppositions entre la réorganisation générale des entreprises et la réorganisation de leurs moyens de production.

Cette thèse s'intéresse au pilotage des installations de traitement de surface. Ces ateliers ont pour objectif de modifier les caractéristiques physico-chimiques des pièces par immersions successives dans différentes solutions chimiques. Les opérations de traitement sont soumises à des contraintes temporelles fortes (durée de production bornée, interrompue par l'extraction de la pièce par un transporteur), contrairement à l'ordonnancement classique qui est généralement soumis uniquement à des durées de production minimales. Le chapitre 2 de cette thèse s'attachera à détailler ce type d'installations, les contraintes qui lui sont propres, et les méthodes de résolution actuellement exploitées.

Les chapitres 3, 4 et 5 traitent de trois aspects liés à ce type d'unités de production. Le chapitre 3 aborde une double problématique. Une problématique scientifique : est-il possible de piloter de façon purement réactive une installation également contrainte temporellement ? Et une problématique industrielle : comment augmenter la productivité tout en conservant une bonne qualité ? Ce chapitre présente l'antagonisme de ces deux problématiques et propose une solution de couplage scientifiquement novatrice et industriellement adoptée.

Le chapitre 4 présente une étude sur les gains possibles dans ces unités de productions en modifiant leur implantation. Un algorithme génétique permet de déterminer de façon systématique une topologie optimale.

Enfin, le chapitre 5 présente une méthode d'optimisation dans le cas d'un pilotage purement réactif. L'étude présente un algorithme évolutionniste de détermination de l'ordre de lancement des produits dans le système afin de permettre une bonne performance de celui-ci.

Dans cette thèse, nous avons voulu relever le défi de l'utilisation, au niveau industriel, d'une méthode de pilotage novatrice, couplant une méthode d'ordonnancement réactive à une méthode d'ordonnancement prédictive. Le déploiement et l'utilisation permanente de ce mode de pilotage nous a permis de valider cette thèse, répondant ainsi à la double problématique (problématique scientifique de la faisabilité, et problématique industrielle de l'exploitation).

Ces travaux ont régulièrement fait l'objet de publications dans des congrès nationaux et internationaux. Ainsi, nous avons présenté l'utilisation du couplage entre les deux méthodes d'ordonnancement pour un congrès international francophone dans [Chové E. *et al.*, 2009a] et pour un congrès international dans [Chové E. *et al.*, 2009b]. Le chapitre 4 a fait l'objet d'une publication internationale dans [Chové E. *et al.*, 2010]. Ces travaux ont également été soumis dans une revue en cours de relecture par le comité scientifique (dans *Journal of Intelligent Manufacturing*). Enfin, ces années de préparation du doctorat ont été ponctuées par la publications des travaux de master dans [Cardin O. *et al.*, 2008a] et [Cardin O. *et al.*, 2008b] puis en revue dans [Cardin O. *et al.*, 2009].

Chapitre 1 : Évolution du pilotage des unités de production

| | |
|---|-----------|
| 1. Évolution générale des systèmes organisés | 10 |
| 1.1. D'une organisation hiérarchique... .. | 10 |
| 1.2. ... à une organisation répartie..... | 11 |
| 1.3. Synthèse..... | 14 |
| 2. Évolution des systèmes de production | 14 |
| 3. Évolution des systèmes de pilotage..... | 17 |
| 3.1. Pilotage prédictif..... | 19 |
| 3.2. Pilotage dynamique | 20 |
| 3.3. Pilotage réactif..... | 23 |
| 3.4. Comparaison des méthodes de pilotage | 25 |

Les travaux de cette thèse ont pour principal objectif de proposer un nouveau mode de pilotage pour les installations de traitement de surface, et d'étudier quelques problèmes liés à la conception et au pilotage de ces unités.

Ce premier chapitre présente le contexte actuel des entreprises. Après avoir déployé, avec l'arrivée de l'informatique industrielle, des structures de pilotage régies par un unique centre de décision, les entreprises changent d'orientation. L'augmentation de la complexité du flux des produits et d'informations rend les systèmes centraux complexes. Les entreprises s'orientent alors vers le déploiement d'une multitude de petits centres de décision plus simples et réactifs : c'est ce que l'on appelle le pilotage réparti.

Nous présenterons dans un premier temps les différentes organisations possibles, en partant des systèmes traditionnels centralisés pour aller jusqu'aux systèmes privés de hiérarchie. Nous aborderons ensuite l'application de cette mutation aux systèmes de production qui passent d'une structure figée pour évoluer vers une structure reconfigurable, voire ré-implantable. Nous terminerons ce chapitre par l'évocation des modifications des systèmes de pilotage d'un mode centralisé, prédictif, optimisé au niveau global, vers un pilotage réparti, réactif, flexible.

1. Évolution générale des systèmes organisés

1.1. D'une organisation hiérarchique...

Dans le contexte de forte évolution de l'organisation des entreprises, notamment à travers une délocalisation et l'ouverture des marchés à la concurrence, les entreprises ont décidé d'adopter des structures hiérarchisées pour contrôler au mieux leur développement. C'est dans ce cadre économique de la deuxième moitié du XX^{ème} siècle que le concept de *Computer Integrated Manufacturing (CIM)*, normalisé par la suite dans [IEC 61512, 1997], a permis de décrire l'organisation des différents systèmes (entreprise, site, zone, cellule de processus, unité, module d'équipement, module de commande). Cet agencement est couramment représentée sous forme d'une pyramide étagée correspondant aux différents niveaux hiérarchiques du système modélisé.

On retrouve deux difficultés pour passer du système à modéliser à une modélisation CIM soulignées dans [Brandimarte P. et al., 1995] :

- *Qu'est ce que la modélisation ?*
- *Comment modéliser ?*
 - Que voulons-nous modéliser ?
 - Pourquoi voulons-nous un modèle ?
 - Quel degré de formalité est nécessaire ?
 - Combien de personnes sont impliquées dans le processus de modélisation ?
 - Quelle facilité de compréhension du modèle est souhaitée, et quelle aide est fournie ?
 - Quelles sont les données nécessaires à l'approche de modélisation utilisée ?
 - Quelles erreurs vont être introduites par le processus de modélisation ?
 - Quelle est la durée de vie du modèle ?

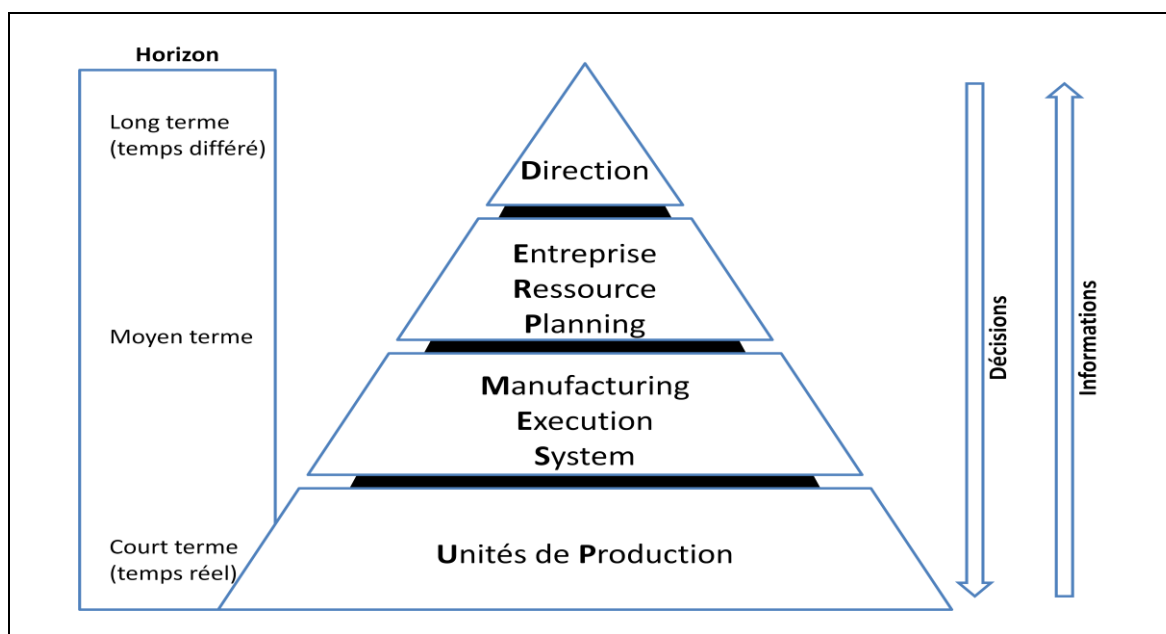


Figure 1 : Exemple de pyramide CIM d'un site de production

La représentation d'une structure hiérarchique sous forme d'une pyramide CIM repose sur des principes constants représentés dans la figure 1.

En partant de ce principe organisationnel, on retrouve une littérature abondante appliquée à différents niveaux de modélisation. Waldner propose dans l'ouvrage [Waldner J.-B., 1990] d'appliquer cette modélisation hiérarchique à l'entreprise. On retrouve sa modélisation dans la figure 2. Le site de production de la figure 1 serait alors une partie du niveau inférieur de l'entreprise de la figure 2.

Ce principe d'organisation montre rapidement des limites dans un monde où la personnalisation des produits est le maître mot pour satisfaire les clients, et la réactivité, un outil indispensable à la compétitivité. Le processus de prise de décision appliqué dans ces structures hiérarchisées allonge le temps de réponse à la demande furtive des marchés.

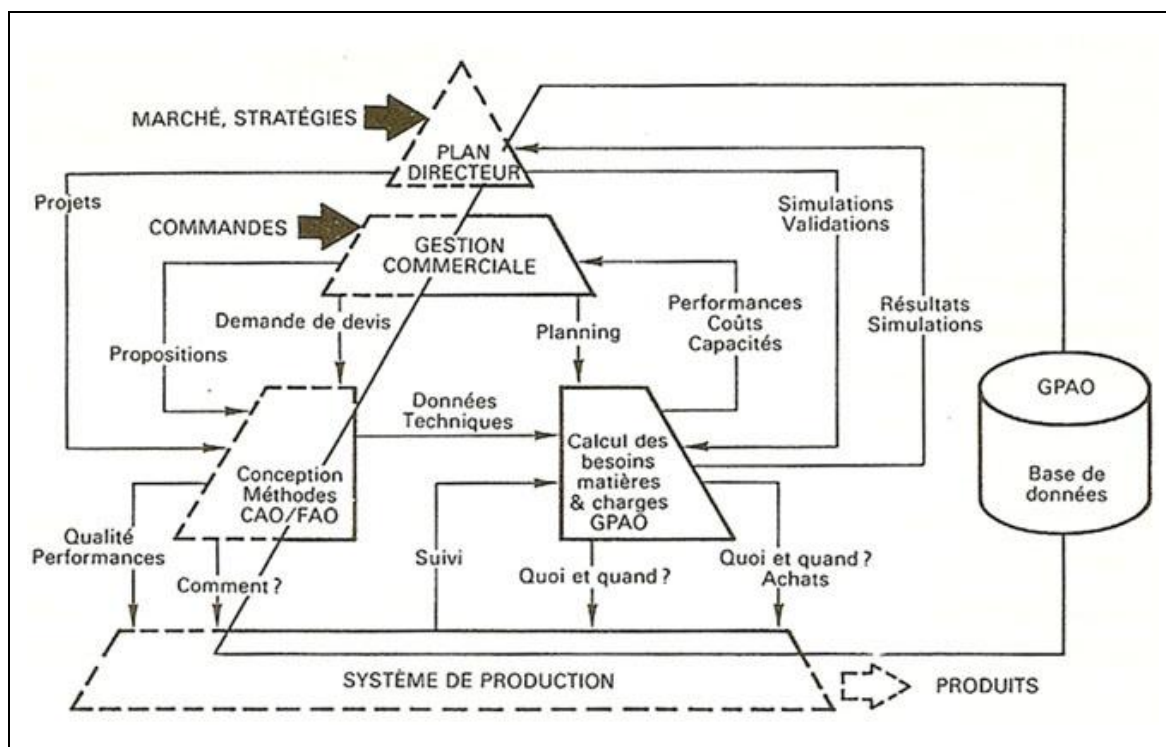


Figure 2 : Pyramide CIM appliquée au niveau de l'entreprise

1.2. ... à une organisation répartie

Les impératifs de réactivité recherchés par les entreprises et les sites de production ont conduit à l'étude de la réduction des délais de prise de décision. Cette recherche de réactivité a, peu à peu, mené à des prises de décision au niveau le plus bas possible.

L'organisation en étoile de l'entreprise (autour d'un centre hiérarchique) a laissé la place à une organisation maillée, où chaque nœud prend des décisions locales sans avoir besoin d'en référer au niveau hiérarchique supérieur. Cette délégation de la prise de décision effectuée par les niveaux hiérarchiquement supérieurs permet une meilleure réponse à la demande et aux aléas, donc une meilleure réactivité.

On retrouve désormais des structures de pilotage plus ou moins réparties. Nous présenterons ici quelques modélisations de systèmes répartis.

1.2.1. Organisation hétérarchique

L'organisation d'un système repose sur deux dimensions de l'organisation : une dimension horizontale et une dimension verticale. Damien Trentesaux propose, dans [Trentesaux D., 2002], de définir ces deux niveaux comme suit :

- Le mécanisme d'agencement vertical est lié au principe de hiérarchie (présenté au paragraphe 1.1). Le fonctionnement d'un niveau et sa performance sont liés au fonctionnement et à la performance des niveaux inférieurs. Il aura alors le contrôle des niveaux inférieurs et en recevra les comptes-rendus d'exécution.
- Le mécanisme d'agencement horizontal est lié à l'absence de hiérarchie entre les entités. Cet agencement est qualifié, par l'auteur, de structure *hétérarchique*. Dans une telle structure, les entités se trouvent au même niveau hiérarchique et peuvent interagir entre elles.

Ces deux mécanismes d'agencement sont décrits par l'auteur dans la figure 3.

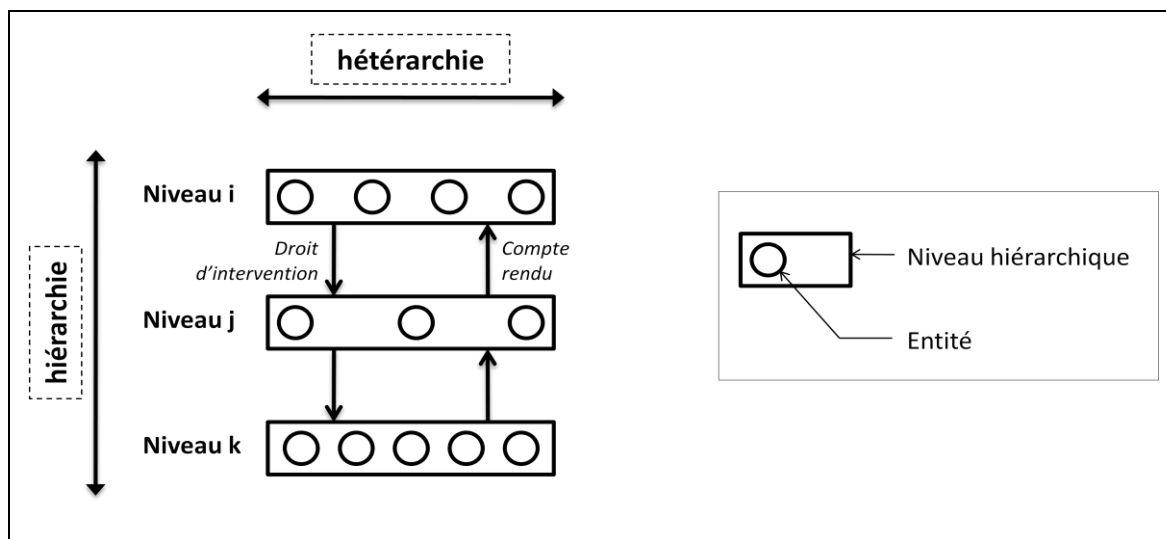


Figure 3 : Les deux mécanismes d'agencement : hiérarchie et hétérarchie

On retrouve des systèmes purement verticaux, horizontaux, et des systèmes couplant hiérarchie et hétérarchie. L'auteur présente ensuite plusieurs classifications plus détaillées des problèmes basés sur ce principe de double orientation.

1.2.2. Organisation holonique

Le concept de Holon a été introduit par le philosophe hongrois Arthur Koestler en 1969 dans [Koestler A., 1969]. Il se base sur le fait qu'une entité peut être vue comme un 'tout' constitué de sous-entités, ou comme une 'partie' constitutive d'une entité plus grande. Le principe organisationnel et interactionnel dans une holoarchie issue de [Müller J.-P. et al., 2005] est présenté dans la figure 4.

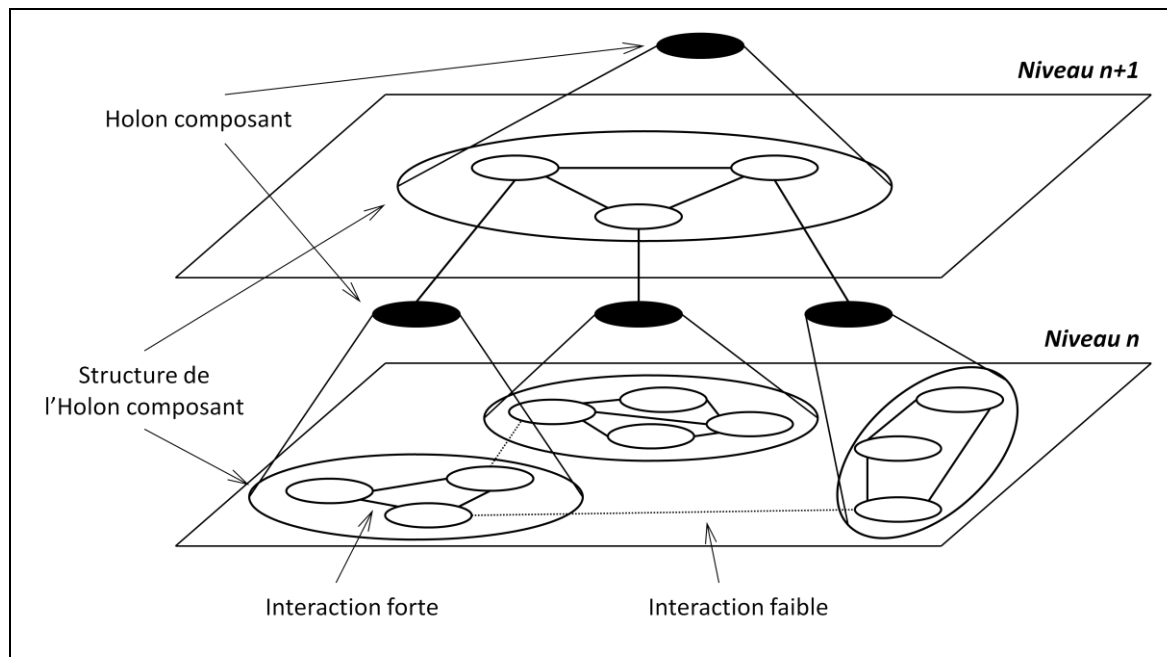


Figure 4 : Organisation d'une holoarchie

Cette organisation est adaptée aux systèmes de production hiérarchisés où, d'une part, chaque sous système doit disposer d'une certaine autonomie et, où d'autre part, l'ensemble de l'organisation vise des objectifs communs (satisfaction des clients, productivité, rentabilité...).

L'architecture holonique de référence PROSA (*Product – Ressource – Order – Staff Architecture*) a été développée dans [Van Brussel H. *et al.*, 1998]. Elle propose une modélisation à partir de quatre Holons de base : Holons produits, Holons ressources, Holons ordres, Holons staff. Pascal Blanc a proposé dans [Blanc P., 2006] une application industrielle d'un système de pilotage basé sur cette architecture. Il a développé différentes heuristiques (fonctionnement et communication) implantées dans chaque Holon. Ces mécanismes, testés dans un système multi-agent, ont permis de montrer la faisabilité d'un système de pilotage basé sur cette organisation. Les performances de ce système ont été étudiées par simulation, dont les résultats sont présentés dans ce rapport.

1.2.3. Organisation isoarchique

La vision extrême de l'organisation répartie est l'isoarchie. Dans une organisation isoarchique, aucun centre décisionnel n'a de pouvoir supérieur à un autre centre. Le système devient alors auto-organisé afin de faire émerger un comportement cohérent. Cette organisation se trouve, sur l'échiquier de la structure architecturale, à l'opposé de la centralisation. Ce concept présenté dans [Pujo P. *et al.*, 2008] se retrouve appliqué aux systèmes holoniques à travers la '*Flat Holonic Form*' présenté dans [Bongaerts L. *et al.*, 2000], repris dans la figure 5. Dans cette organisation, le système est composé d'Holons de même niveau, sans possibilité d'existence d'un *Holon Staff*.

Cette organisation peut se rapprocher du fonctionnement d'une fourmilière, et des algorithmes bio-inspirés qui en découlent. Chaque Holon représente une fourmi qui interagit avec ses voisins afin de faire émerger un comportement évolutionniste cohérent pour l'obtention de l'objectif souhaité.

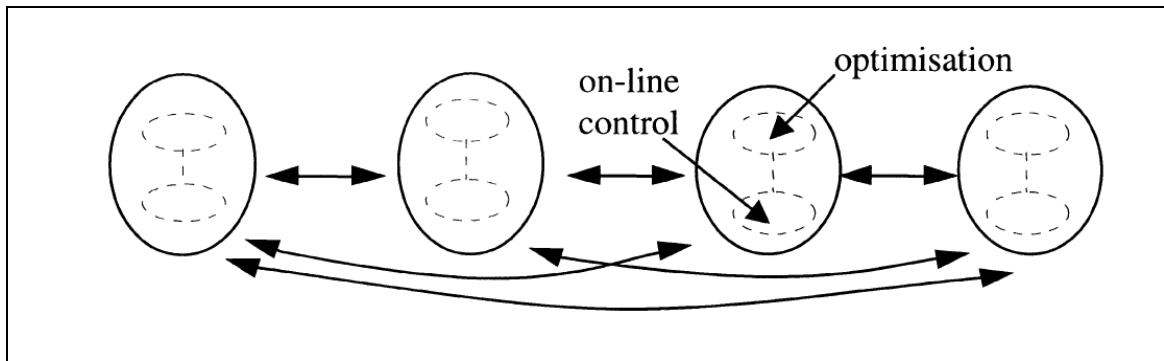


Figure 5 : Flat Holonic Form

1.3. Synthèse

On trouve ainsi, dans le contexte économique en pleine mutation, une modification des structures de pilotage. Elles laissent de plus en plus d'autonomie aux niveaux inférieurs en passant d'un contrôle hiérarchique à un contrôle réparti.

La centralisation présente l'avantage d'une vision large du système, afin d'orienter son fonctionnement vers un pilotage optimal, tandis que les systèmes répartis ont l'avantage de mieux répondre aux aléas et à la réorganisation.

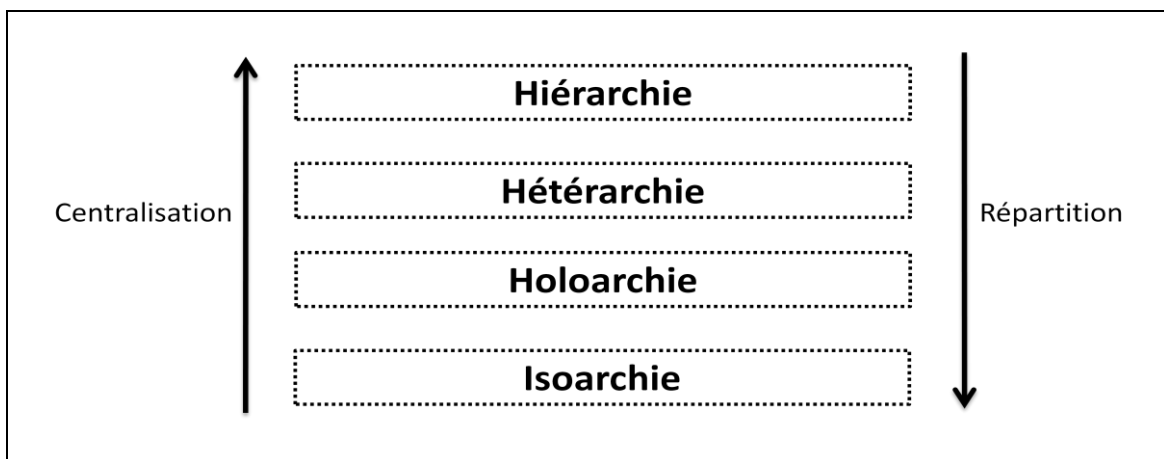


Figure 6 : Évolution des systèmes organisés

2. Évolution des systèmes de production

Les systèmes de production industriels ont, au cours du XX^{ème} siècle, connu une grande évolution. L'industrie des *temps modernes*, présentée par *Sir Charles Spencer Chaplin*, a peu à peu disparue pour laisser place à une nouvelle organisation du travail. Cette application intensive du fordisme (dont le taylorisme est une des composantes) a permis de rationaliser le travail industriel pour produire rapidement et à faible coût. L'inconvénient de cette organisation a été le manque de personnalisation des produits, gros handicap face à une demande de plus en plus diversifiée.

Il a fallu attendre les années 1980 pour répondre à ce manque de personnalisation, avec l'arrivée des *systèmes de production flexibles* ou *flexible manufacturing systems (FMS)*. Cette nouvelle implantation des systèmes de production brevetée par [Johnstone R. et al., 1984] s'organise autour d'un système de convoyage automatisé des produits, desservant un ensemble de machines de production. Les produits entrent dans l'unité de production et sont déposés sur le convoyeur. Acheminés par celui-ci sur les différentes machines, ils vont effectuer une gamme de fabrication qui leur est propre afin de répondre à une demande unique de production.

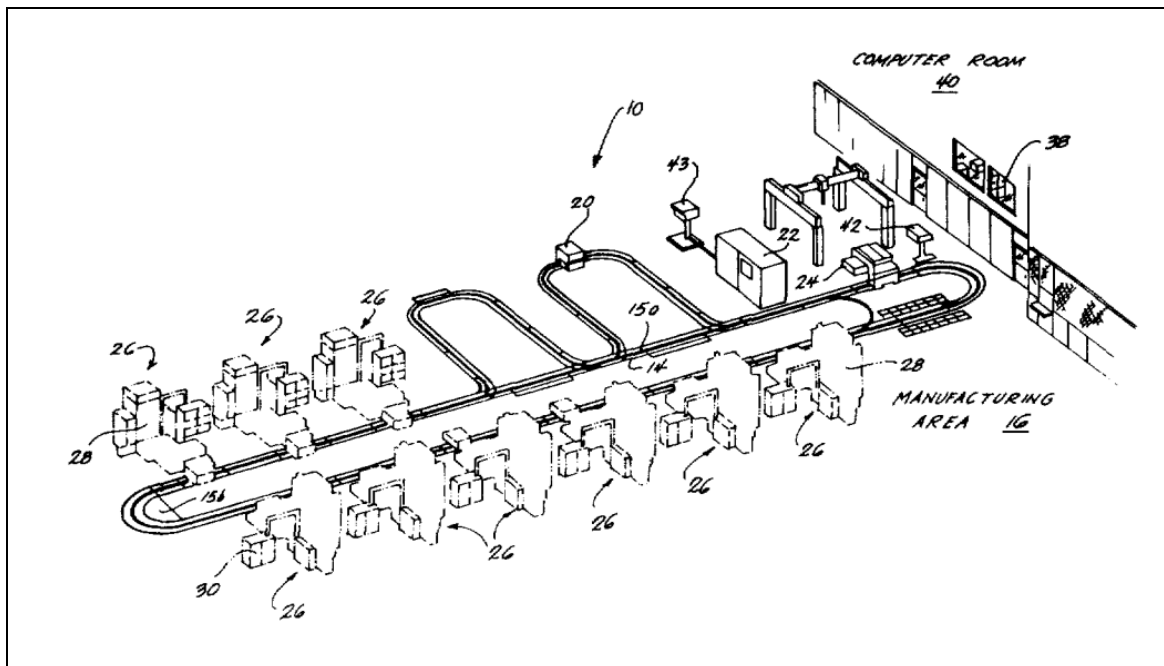


Figure 7 : Exemple de FMS proposé dans le brevet USP 4.472.783

Cette nouvelle organisation des ateliers de production permet de répondre à beaucoup de contraintes liées aux systèmes de production :

- Chaque produit entrant dans le système peut avoir une gamme de fabrication propre pour répondre à la **demande de personnalisation** grandissante des produits. Ce type de production correspond, dans la littérature, à l'appellation de JobShop.
- Un même système de production peut facilement prendre en compte de **nouvelles références** sans que celle-ci aient explicitement été prévues au moment de l'implantation du système.
- Une machine peut être ajoutée ou enlevée du système pour mieux répondre à la **charge de production**.
- Plusieurs machines peuvent être capables de produire les mêmes produits pour sécuriser le système par la **redondance des machines**.
- ...

L'étape suivante, cherchant à augmenter la prise en compte de productions hétéroclites, est la **reconfigurabilité** des systèmes. Ce paradigme récent, présenté dans [Bi Z.-M. et al., 2008], décrit la capacité d'un système de production à se reconfigurer afin de répondre à une augmentation du nombre de produits et une personnalisation croissante. Il n'est plus simplement question de gammes

de fabrications variables, mais d'opérations de productions variables. Dans ce paradigme présenté dans la figure 8, les machines sont évolutives afin de prendre en charge de nouvelles opérations et/ou le pilotage (la politique) de l'entreprise est souple afin de répondre à de nouvelles demandes du marché.

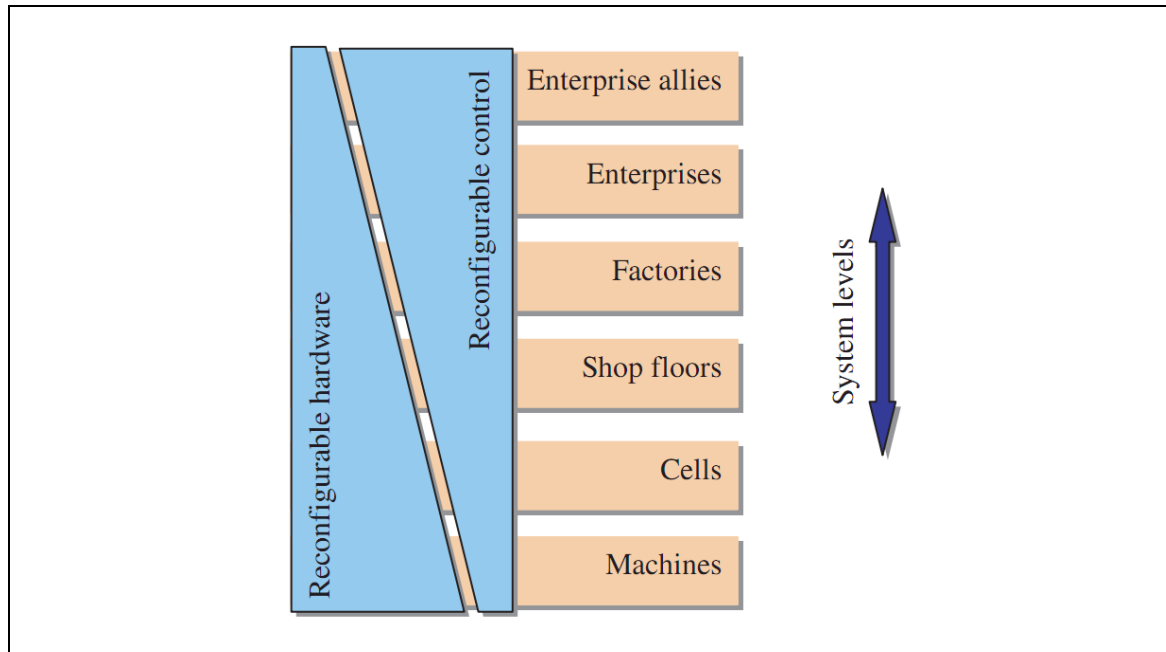


Figure 8 : Organisation d'un système reconfigurable

On retrouve dans l'ouvrage [Mori M. *et al.*, 2009] deux niveaux de reconfigurabilité mettant en œuvre des mécanismes et des niveaux différents :

- La **reconfigurabilité des machines** met en avant la capacité d'une machine à être modifiée pour répondre à de nouvelles opérations à réaliser. Il s'agit ici d'obtenir des machines modulaires évolutives. Berruet a proposé dans [Berruet P., 2007] une méthode de génération automatique de la commande dans de tels systèmes pour permettre une reconfiguration plus aisée de ceux-ci.
- La **reconfigurabilité des systèmes** reflète la capacité d'un système à être réimplanté afin de répondre à de nouvelles gammes.

Toutes ces évolutions ont été favorisées par l'apparition de l'automatisation importante des ateliers et la banalisation des systèmes d'informations au sein des entreprises. Cette nouvelle topologie des systèmes de production soulève néanmoins de nouveaux problèmes tel que les problèmes de dimensionnement d'unité, de choix d'implantation, de routage des produits...

[Bi Z.-M. *et al.*, 2008] classe les paradigmes proposés précédemment, comme des solutions répondant aux quatre objectifs principaux d'une entreprise actuelle :

- **Délai d'exécution court.** Afin de répondre rapidement aux clients et de proposer des produits adaptés avant la concurrence, une entreprise doit avoir un cycle court de développement des produits et une réponse rapide à la demande du marché.

- **Personnalisation.** Les produits étant de plus en plus personnalisés, une entreprise doit être capable de proposer un large choix de variantes pour un même produit. On s'oriente ici vers un produit unique réalisé à la demande du client. Cette contrainte rentre rapidement en conflit avec la précédente et impose une bonne organisation de la production.
- **Demande variable.** Le cycle de vie des produits étant de plus en plus court, et les phénomènes de modes très importants, il faut être capable, sans faire de stock pour ne pas risquer de conserver des invendus, de répondre à une demande fortement variable.
- **Prix économique.** Le client cherchant souvent à maximiser le rapport *qualité / prix*, un des objectifs non négligeable est de baisser les prix, donc de minimiser les coûts de production.

On retrouve les paradigmes cités précédemment, classés suivant ces quatre objectifs, dans [Bi Z.-M. *et al.*, 2008] repris dans la figure 9.

Cette classification permet, à partir des objectifs visés et des stratégies voulues, de choisir le ou les paradigmes de production qui permettront d'obtenir les meilleurs résultats.

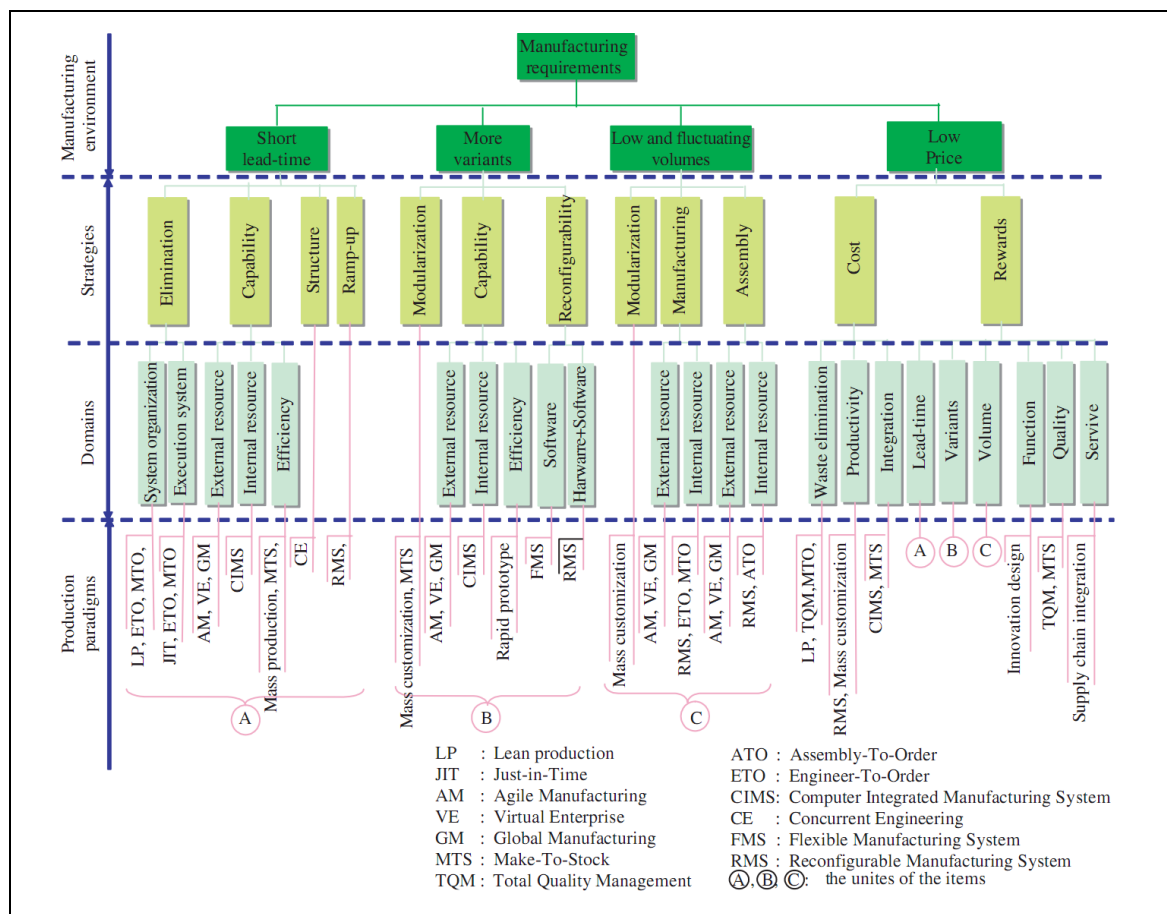


Figure 9 : Paradigmes de productions classés par objectif

3. Évolution des systèmes de pilotage

Tout système (de production ou non) est soumis à des actions extérieures provoquant une organisation et/ou une désorganisation de celui-ci. Certaines de ces actions sont commandées afin

de faire tendre le système vers un état stable. Cette boucle de rétroaction est ce que l'on appelle la boucle de pilotage du système [Sénéchal O., 2004]. Piloter revient donc à fixer un objectif à atteindre, puis à définir cette boucle de rétroaction pour conduire le système dans l'état stable visé.

Dans [Cauvin A., 2005], l'auteur liste les différentes sources de perturbations que la boucle de rétroaction cherche à compenser (voir figure 10) :

- **La disponibilité des ressources.** Ces perturbations sont engendrées par des problèmes de fiabilité (humaine ou des machines).
- **Le défaut d'approvisionnement** dû à des problèmes de fournisseurs, de transporteurs ou des unités amont.
- **Les aléas liés aux clients.** Les changements de quantité de commande, de délai de livraison sont autant de perturbations à résorber.
- **Les défauts d'information.** On regroupe, dans cette classe d'aléa, les problèmes de remontée d'information entre le système piloté et le système de pilotage. Ces défauts peuvent être dus à des problèmes de mesure, de transmission ou d'interprétation de l'information.
- **Les décisions intempestives.** Ces perturbations sont introduites par le pilotage lui-même pouvant résulter de décisions trop rapides ou mal adaptées au système piloté.

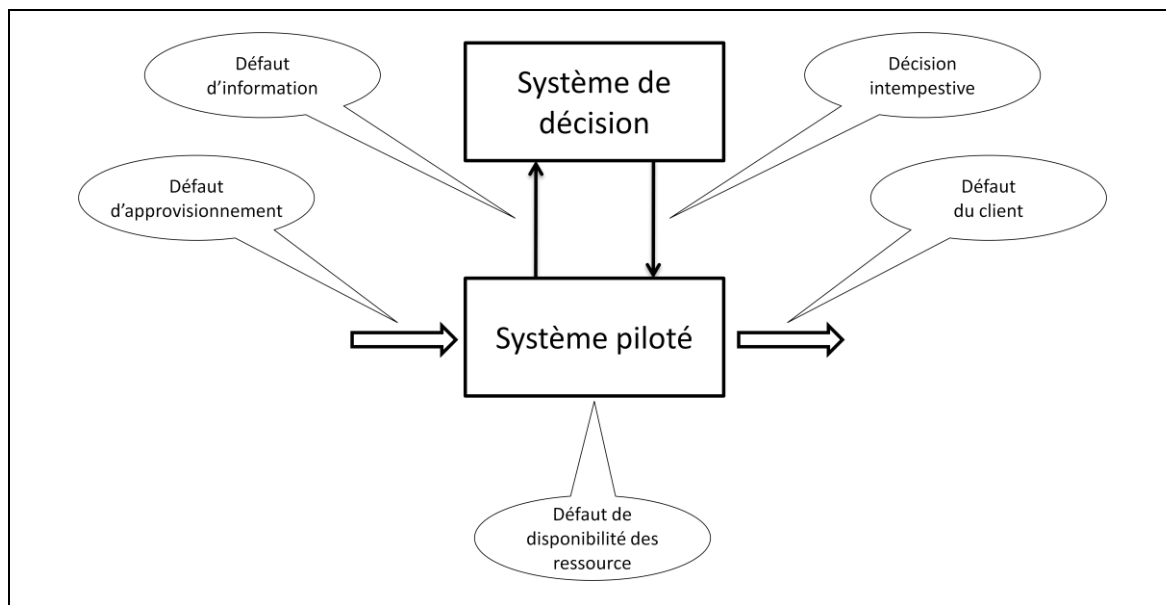


Figure 10 : Sources de perturbations en production selon [Cauvin A., 2005]

Dans [Cardin O., 2007], l'auteur scinde le processus, entre l'occurrence d'une perturbation et la disparition de celle-ci, en 4 étapes : la détection de la perturbation, l'analyse de celle-ci, la conception de la réponse, l'implantation de la réaction. C'est le temps écoulé entre l'occurrence de la perturbation et sa disparition qui caractérisera un système de pilotage.

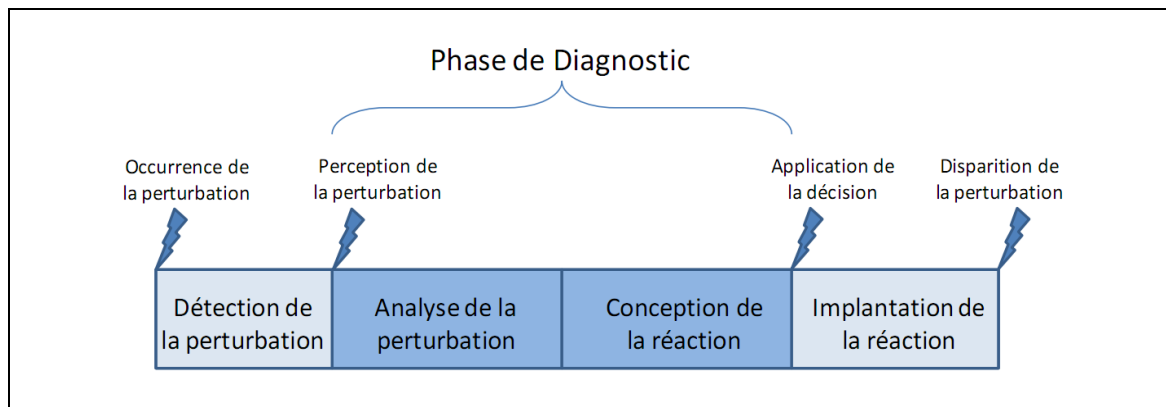


Figure 11 : Cycle de vie d'une perturbation selon [Cardin O., 2007]

La conception du pilotage d'une installation consiste à définir le fonctionnement de la troisième phase du cycle de vie de la perturbation évoquée dans la figure 11. Dans un mode de pilotage prédictif, les perturbations n'engendrent aucune décision tandis que dans un mode de pilotage dynamique ou réactif, il faut prendre intelligemment en charge les perturbations pour maintenir le système opérant dans un état stable.

L'**ordonnancement** de la production consiste, selon [Brucker P., 2006], à allouer à chaque tâche un ou plusieurs intervalles de temps sur une ou plusieurs ressources. L'ordonnancement des tâches peut être défini à l'avance, nous parlerons alors d'ordonnancement prédictif. Dans le cas où cet ordonnancement est construit au fur et à mesure de l'exécution du système, nous parlerons d'ordonnancement réactif. Une troisième méthode consiste à définir l'ordonnancement à l'avance, puis à le remettre en cause en cours d'exécution. Nous parlerons alors d'ordonnancement dynamique. Le **pilotage** consiste ensuite à appliquer une méthode d'ordonnancement (prédictif, réactif, dynamique...) sur une unité de production. Nous utiliserons, dans la suite de ce mémoire, le terme de pilotage prédictif/réactif/dynamique pour évoquer le contrôle d'une unité de production par une méthode d'ordonnancement prédictive/réactive/dynamique.

Les évolutions des systèmes de production, cherchant à répondre à la demande toujours plus personnalisée des clients, ont conduit à revoir les systèmes de pilotage qui les contrôlent. Le pilotage prédictif laisse petit à petit la place au pilotage dynamique et réactif.

3.1. Pilotage prédictif

Dans une **approche prédictive**, l'affectation des tâches sur les différentes ressources est fixée avant le lancement de la production. Dans une telle approche, les résultats sont fournis par un moteur d'ordonnancement et connus à l'avance. Le planning des ressources, les dates prévisionnelles de sortie des produits sont fixés avant le lancement de la production. Cette approche permet d'optimiser sous contrainte l'ordonnancement des tâches pour optimiser la fonction objectif. Elle permet aussi une excellente visibilité de l'évolution future du système, puisque toutes les décisions sont prédites à l'avance. Les calculs étant réalisés a priori, leur temps n'influe pas sur l'exécution du système en production.

Un premier inconvénient d'une telle approche réside dans le retard impliqué par les temps de calculs du moteur d'ordonnancement. La plupart des algorithmes de calcul des ordonnancements sont basés sur des méthodes d'évaluation exhaustive des solutions, conduisant à des temps de calcul augmentant exponentiellement avec la taille des problèmes. Un second inconvénient est que le principe de cette approche ne permet pas de prendre en compte de nouveaux produits ou de réagir aux aléas de production. L'approche prédictive est particulièrement bien adaptée dans des ateliers à forte valeur ajoutée ou dans lesquels on retrouve régulièrement les mêmes séquences de produits à fabriquer.

On retrouve une littérature abondante, durant ces 50 dernières années, sur le problème d'ordonnancement des tâches. L'ouvrage [Pinedo M., 2008] est une synthèse reprenant différents travaux relatifs à l'ordonnancement. L'auteur classe les méthodes de résolution en deux catégories : les modèles déterministes, basés sur le principe de causalité, et les modèles stochastiques, introduisant des probabilités d'occurrence pour certaines décisions.

Gourgand et ses co-auteurs, dans [Gourgand M. *et al.*, 2003] puis dans [Gourgand M. *et al.*, 2005], proposent une méthode d'évaluation du *makespan* dans le cadre d'un flow-show dont les durées de production sont indépendantes les unes des autres et répondent à une loi exponentielle. Cette méthode, basée sur un algorithme itératif utilisant les chaînes de Markov, a ensuite été intégrée dans une méta-heuristique d'ordonnancement (le recuit simulé) afin de déterminer le planning qui minimise le *makespan*. L'intérêt de ces travaux est de déterminer, à l'avance, un ordonnancement optimisé dans lequel les durées de production ne sont pas connues exactement. Par conséquent, l'ordonnancement répond à un objectif de stabilité du système et d'optimisation du moyen de production, dans le cas de durée aléatoires. Ces objectifs souvent nécessaires sont régulièrement absents des ordonnancements prédictifs.

3.2. Pilotage dynamique

Le **pilotage dynamique** est une amélioration du pilotage prédictif dans un environnement non prévisible. A partir d'un encours de production et d'un lot de produits à réaliser, il produit un nouveau planning de production. Il peut alors être exécuté pendant la production afin de prendre en compte un aléa ou une nouvelle commande.

Le pilotage dynamique fait partie des outils de pilotage *en ligne*. Il a le double avantage de réagir aux aléas survenus pendant la production et de permettre d'optimiser le planning avant de l'exécuter. Dans [Sgall J., 1998], l'auteur souligne que l'ordonnancement en ligne reflète la problématique industrielle réelle. Il rappelle que les travaux sur la complexité des algorithmes d'ordonnancement, notamment la NP-complétude de beaucoup de méthodes d'ordonnancement, ont rendu nécessaire le développement d'heuristiques exécutables en temps réel.

La différence entre le pilotage prédictif et le pilotage dynamique réside principalement dans les algorithmes d'ordonnancement utilisés. Beaucoup de travaux cherchent à développer des algorithmes et des méta-heuristiques fournissant un résultat en un *temps acceptable*. Ces méthodes, basées sur une exploration non exhaustive du champ des solutions, permettent d'obtenir

rapidement une solution pour répondre au problème du pilotage dynamique. Cependant cette solution n'est pas (en général) la solution optimale au problème.

La figure 12 permet de repérer deux types de phases pendant l'exécution d'un système piloté dynamiquement :

- **Phases d'exécution normale (A & C).** Dans ces phases, le système de production exécute le planning proposé par l'ordonnanceur. Cette phase dure jusqu'à ce qu'un nouvel ordonnancement soit demandé au moteur d'ordonnancement.
- **Phases d'exécutions dégradées (B & D).** Dans ces phases allant de t_{demande} à $t_{\text{réponse}}$, il peut être décidé d'arrêter la production ou de la continuer. *Continuer la production* permet d'exploiter le moyen de production pendant l'ordonnancement mais peut introduire une incompatibilité entre l'état réel du système à $t_{\text{réponse}}$ et le planning proposé par le moteur d'ordonnancement, dégradant la stabilité du système. *Arrêter la production* dégrade les performances du système de production, et dans certains cas peut dégrader la qualité de la production (par exemple dans le cas d'unités de traitement de surface, d'unités de production alimentaire, de processus faisant intervenir des chauffages et refroidissements).

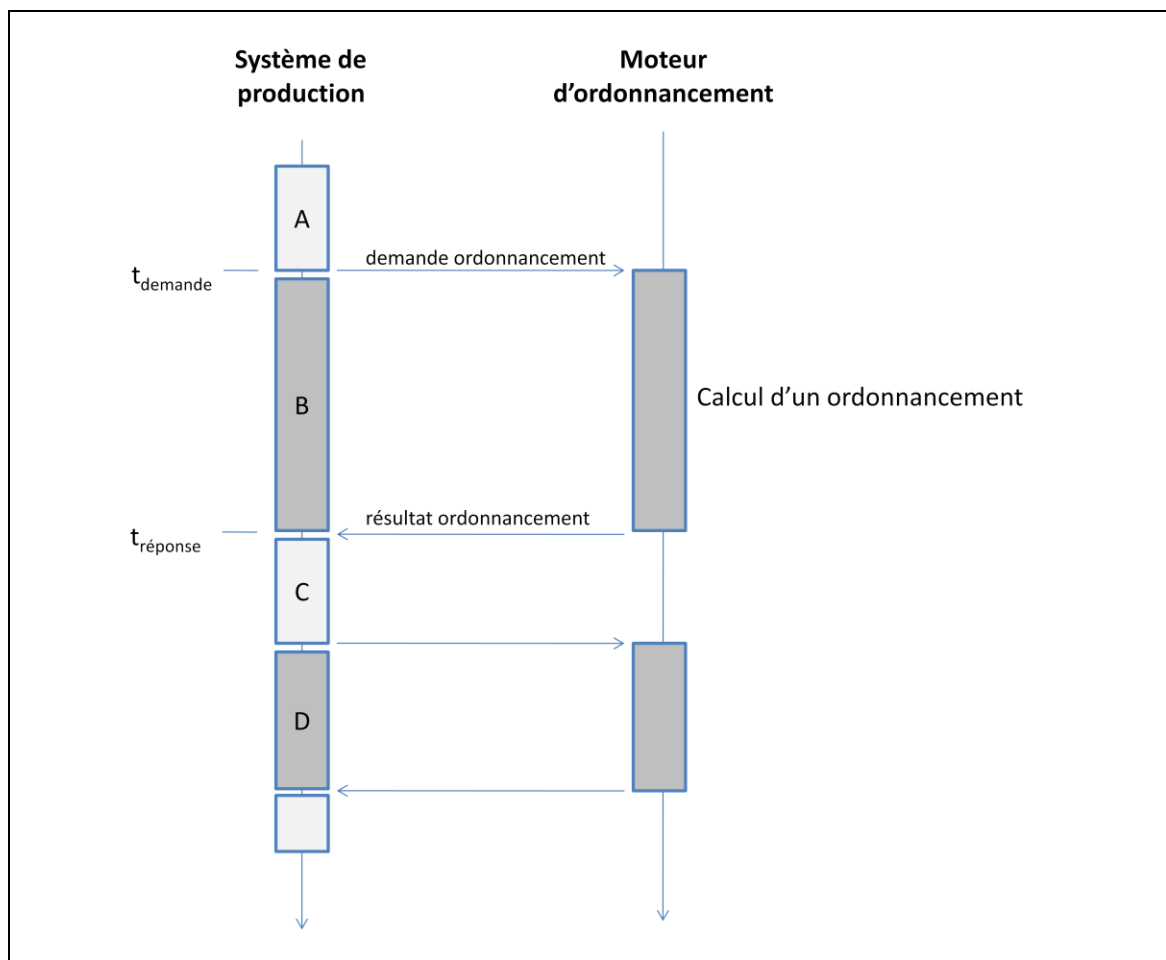


Figure 12 : Pilotage dynamique d'un système

Les auteurs de [Pfeiffer A. et al., 2007] ont réalisé une classification des méthodes de résolution du problème de ré-ordonnancement. Si l'on considère l'application à un cas réel, il ne faut pas

demander inutilement des ré-ordonnancements car ils perturbent le système et risquent de dégrader ses performances. Les auteurs classent les stratégies de demandes de ré-ordonnancement en trois catégories :

- Ré-ordonnancement périodique
- Ré-ordonnancement déclenché par événement
- Stratégie hybride

Les auteurs proposent une architecture de tests pour évaluer les performances du couplage entre un système de production et son moteur de ré-ordonnancement en ligne. Cette architecture se base sur l'émulation du système à l'aide d'un simulateur temps réel introduisant des perturbations dans l'exécution du planning. Les résultats des tests effectués relatent l'influence des stratégies de re-planification sur la stabilité du système.

Enfin, les auteurs soulignent l'inutilité de planifier à long terme, car une re-planification remet souvent en cause tout le planning préétabli. Les opérations à moyen ou long terme ont une forte probabilité d'être modifiées par une future planification. Le temps passé à planifier sur une longue période aura alors été perdu.

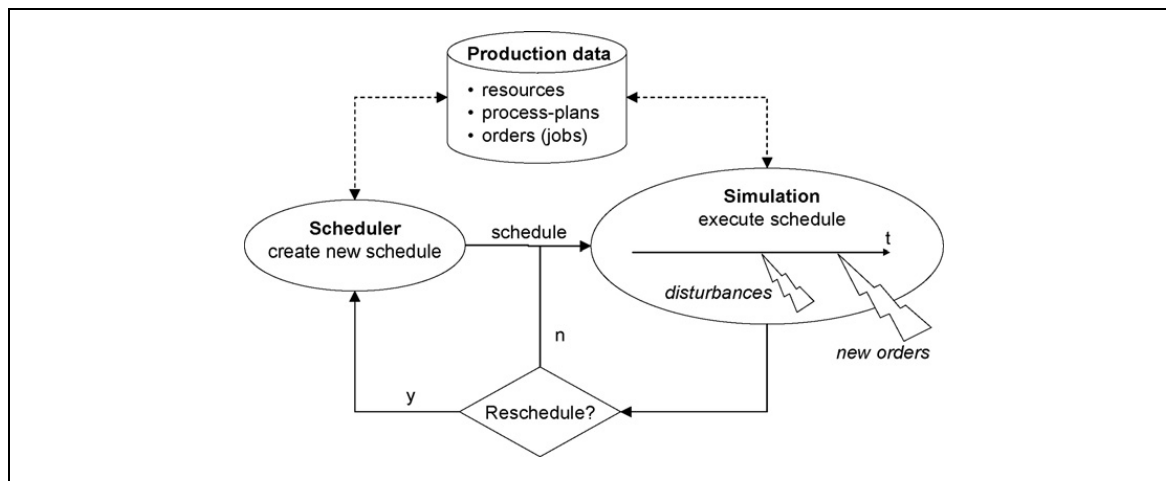


Figure 13 : Architecture d'évaluation d'un ré-ordonnanceur

Dans [Zhao C. *et al.*, 2010], les auteurs présentent une application du problème de **ré-ordonnancement** d'atelier en environnement perturbé. Contrairement à beaucoup d'autres travaux sur les ré-ordonnancement en ligne, les auteurs insistent sur l'utilité d'utiliser certaines optimisations réalisées dans le planning précédent. Le *principe d'apprentissage* est bien connu, mais les auteurs proposent ici une mémoire à court terme : on n'utilise que l'occurrence précédente du calcul. Chuanli Zhao et son co-auteur proposent un algorithme polymodal de résolution du problème de ré-ordonnancement avec détérioration. L'algorithme proposé utilise les *optimisations déjà effectuées* dans l'ordonnancement précédent et prend en charge les aléas et l'arrivée des nouveaux produits. Les hypothèses utilisées dans l'algorithme (en particulier le lemme 3), interdisent l'inactivité d'un produit. Cette contrainte rend l'algorithme inapplicable dans beaucoup de cas, notamment pour l'ordonnancement d'ateliers de traitement de surface.

3.3. Pilotage réactif

Dans le cas d'unités où la production à réaliser n'est pas connue à l'avance, ou dans le cas d'unités ayant de nombreux phénomènes aléatoires ou non prévisibles⁵, la prévision proposée par le pilotage dynamique sera très souvent remise en cause. Dans ce cas, les phases dégradées prennent une part importante du temps de la production. Dans [Herroelen W. *et al.*, 2004], les auteurs ont proposé de réaliser un inventaire des méthodes réactives de planification de projet. Les méthodes sont comparées en fonction de leur robustesse et de la stabilité du planning généré. Cette même robustesse et la stabilité vis-à-vis des perturbations de l'atelier ont été étudiées à l'aide d'une méthode floue dans [Dorn J. *et al.*, 1995].

Le **pilotage réactif** cherche résoudre ces problèmes en introduisant un nouveau paradigme de pilotage. Il se base sur l'édiction de règles simples appliquées en temps réel dans chaque centre décisionnel. Dans ce cas, aucune prévision n'est réalisée. Les règles de pilotage introduites dans le pilotage réactif sont simples (complexité de l'ordre 1 ou n), et sont réparties sur tout le système.

Si l'on prend l'exemple d'un atelier de production à deux machines représenté dans la figure 14, on peut dénombrer six centres de décision élémentaires :

- Le répartiteur choisit la machine sur laquelle il envoie les produits.
- Les stocks réorganisent l'ordre des produits.
- L'expédition choisit la source des produits expédiés.

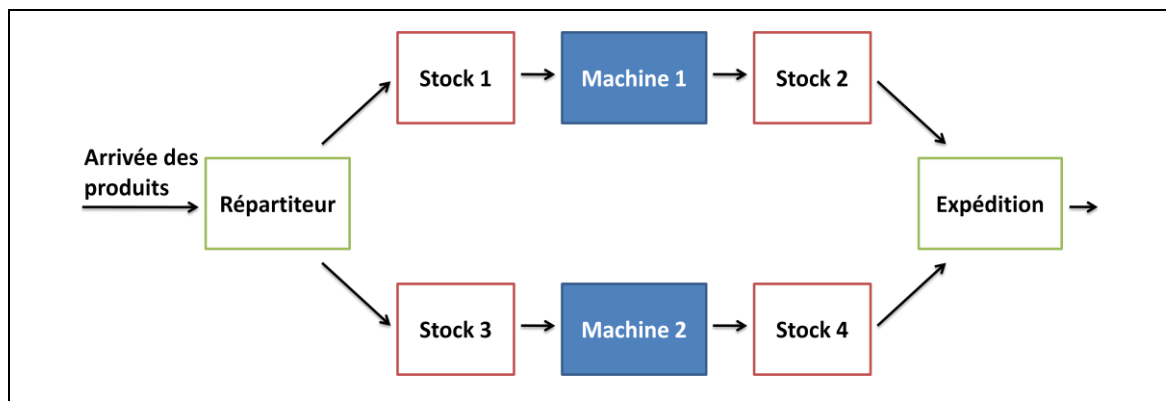


Figure 14 : Atelier de production à deux machines

Les choix pris dans les centres de décision peuvent être établis par un planning prédictif (a priori). Dans ce cas, on peut choisir un critère à optimiser : équilibre de la charge des postes, minimisation du nombre de réglages, respect des dates de livraison... Cet atelier peut aussi être piloté de façon réactive en implémentant une règle locale dans chaque nœud de décision. Chaque règle n'a qu'une vision restreinte du problème. Par exemple, le répartiteur peut décider d'envoyer deux produits sur la machine 1, puis un produit sur la machine 2, ou de choisir la machine ayant le moins de produits dans son stock d'entrée.

⁵ Par exemple dans les *systèmes flexibles de production* où les temps de transfert sont variables et difficilement prévisibles, ou pour des tâches à durée non définies a priori.

La **difficulté de l'ordonnancement réactif** réside dans la définition des règles de pilotage. Elles doivent, par leur action locale, concourir à l'émergence d'un comportement global cohérent et répondant aux objectifs de l'unité de production. Il est possible de passer de la définition locale des règles à une approche du **comportement émergent** par simulation du système réel muni des règles de pilotage. Le contraire n'est pas possible : à partir d'un comportement global souhaité, on ne peut obtenir, de façon systématique, la définition des règles locales.

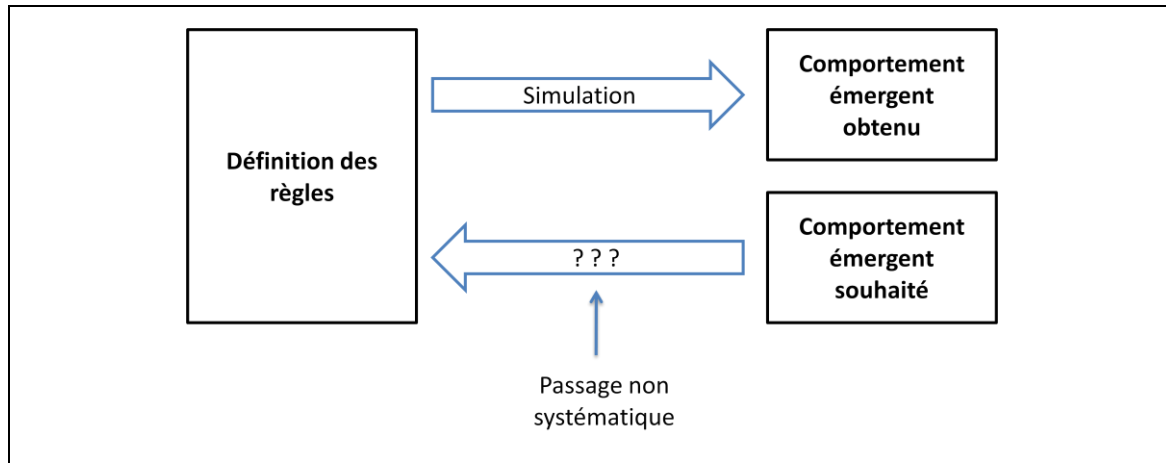


Figure 15 : Lien entre règles locales et comportement global

La **définition des règles** locales se fait donc par approche évolutionniste manuelle. Les règles sont ajoutées successivement au système. Elles sont ensuite simulées afin de déterminer le comportement émergent, puis l'adéquation entre le comportement souhaité et le comportement obtenu. Une analyse de cette adéquation permet de compléter la liste des règles. La simulation n'étant pas une preuve, et étant faite sur un jeu spécifique de données, il est alors très difficile, voire impossible, de garantir la cohérence des règles dans tous les cas.

Le pilotage réactif est particulièrement adapté aux systèmes répartis (voir paragraphe 1.2) dans lesquels les centres de décisions sont situés au plus près des actionneurs qu'ils commandent. Dans ce cas, les règles peuvent être implantées localement, dans les bas niveaux hiérarchiques. On obtient alors une application du paradigme informatique des **systèmes multi-agents**. Dans ce cas il est souhaitable que le produit porte l'information qui s'y rapporte afin de déclencher les décisions qui le concernent, ou que le produit détienne l'intelligence pour prendre ces mêmes décisions. Cette application du pilotage réactif, appelé **pilotage par le produit** est présenté dans [Pannequin R., 2007]. Le produit acquiert alors une certaine *intelligence* que [Cea Ramirez A.-A., 2006] classe suivant quatre types en fonction des capacités de l'objet :

- Produit porteur de données
- Produit pointeur vers un système d'information
- Produit fournisseur et demandeur de services
- Produit sensible

Les règles définies dans le pilotage réactif sont alors déclenchées par les événements des produits, et les décisions sont prises par le centre d'action ou par le produit lui-même.

Enfin, la dernière étape liée à la flexibilité des systèmes de pilotage est la **reconfigurabilité** de ceux-ci. La reconfigurabilité des systèmes de pilotage est liée à la reconfigurabilité des systèmes de production abordée dans le paragraphe 2. Dans [Mehrabi M.-G. *et al.*, 2000], les auteurs définissent un système reconfigurable comme un système flexible, non seulement dans sa capacité à produire une grande variété de pièces, mais aussi dans sa capacité à se modifier lui-même. Pour ce faire, le système est créé comme un agrégat de modules matériels et logiciels assemblés entre eux. Ce principe réduit le risque d'obsolescence du système en lui permettant de s'adapter continuellement à l'évolution des demandes. Le pilotage réparti est particulièrement bien adapté à ces systèmes, introduisant une bijection entre les modules matériels et les modules logiciels.

3.4. Comparaison des méthodes de pilotage

Nous avons vu qu'il existait trois grandes classes pour la résolution du problème d'ordonnancement. J. Sun et D. Xue, dans [Sun J. *et al.*, 2001], constatent que la plupart des algorithmes d'ordonnancement, et des systèmes de productions qui leur sont associés, sont liés au problème de l'ordonnancement prédictif. Ces algorithmes ont des difficultés pour résoudre les problèmes industriels actuels : « *Premièrement, le problème d'ordonnancement d'atelier est dynamique par nature, c'est-à-dire que de nouveaux ordres de fabrication peuvent arriver continuellement pendant la production. Deuxièmement, le planning proposé peut être changé pour refléter les changements dans les ordres de fabrication et dans les conditions de production.* ». Dans le cas des algorithmes d'optimisation, c'est souvent la recherche d'optimalité qui prime sur la possibilité d'une exécution en temps réel.

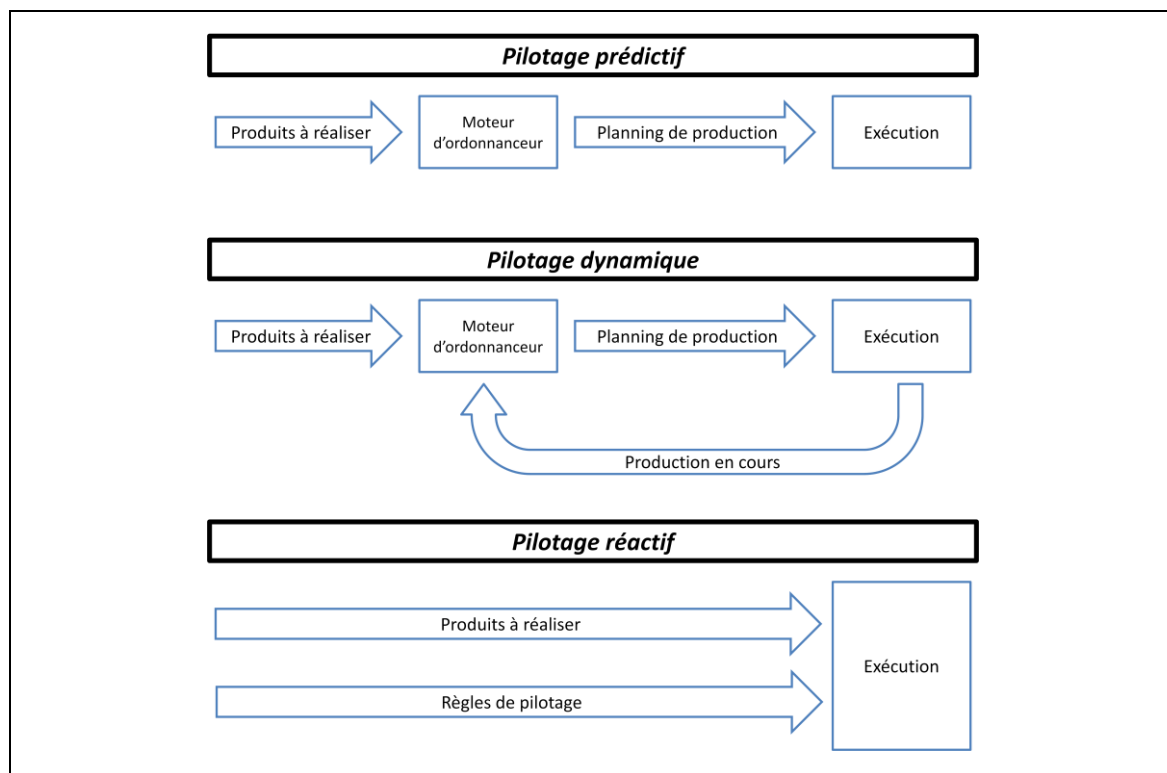


Figure 16 : Principe de fonctionnement des trois classes de pilotage

Les trois classes de pilotage peuvent être synthétisées par leur principe de fonctionnement rapporté en figure 16. L'étude faite dans les paragraphes précédents nous a permis de déterminer le tableau de la figure 17.

L'ordonnancement prédictif, selon [Van de Vonder S. *et al.*, 2007], a suscité de nombreuses recherches au cours de la dernière décennie. Ce type d'ordonnancement étant applicable dans le cadre d'un environnement **omniscient** (toute l'information est disponible), **statique** (les données ne changeront pas avec le temps) et **prédictif** (le système réagira tel que souhaité). Dans un tel cadre, le pilotage prédictif basé sur un ordonnancement optimisé est idéal pour obtenir les meilleures performances. Dès que l'on s'écarte d'un tel cadre, et que l'on souhaite être réactif et obtenir un ordonnancement robuste, il faut s'orienter vers un **ordonnancement réactif**.

| Critère | Pilotage Prédictif | Pilotage dynamique | Pilotage réactif |
|-------------------------------------|---|---|--|
| Mode de planification | Hors ligne | En ligne | En ligne |
| Prise en compte nouvelle production | Non | Oui | Oui |
| Horizon de prévision | Long | | Nul |
| Possibilité d'optimisation | Oui | Oui sous contrainte temporelle | Faible |
| Réaction aux aléas | Nulle | Moyenne | Forte |
| Complexité | Complexité des algorithmes mis en œuvre | Dilemme entre performance du résultat et du temps de calcul court | Difficulté de la définition des règles Comportement émergeant incertain |

Figure 17 : Comparaison des trois classes de pilotage

Chapitre 2 : Ordonnancement des installations de traitement de surface

| | |
|--|-----------|
| 1. Description des unités de traitement de surface..... | 28 |
| 1.1. Les porteurs | 28 |
| 1.2. Les cuves | 29 |
| 1.3. Les transporteurs (ponts roulants) | 31 |
| 1.4. Les navettes | 34 |
| 2. Problématique du Hoist Scheduling Problem..... | 35 |
| 2.1. Définition d'une gamme de traitement..... | 35 |
| 2.2. Notations..... | 36 |
| 2.3. Contraintes d'ordonnancement | 38 |
| 2.4. Fonction objectif | 39 |
| 2.5. Intégration du HSP dans les problèmes d'ordonnancement..... | 40 |
| 3. Classification des problèmes | 42 |
| 4. Approches de résolution | 45 |
| 4.1. Méthodes hors ligne | 45 |
| 4.2. Méthodes en ligne | 50 |
| 4.3. Bilan sur les méthodes existantes..... | 53 |
| 5. Conclusion..... | 53 |

Un **traitement de surface** est une opération chimique, électrochimique ou mécanique qui a pour objectif de modifier les caractéristiques de la surface d'une pièce : nettoyage, modification de la dureté, décoration, anticorrosion...

Une **ligne de traitement de surface** est une unité de production capable de réaliser différentes opérations de traitement de surface. Une installation est constituée de postes de traitement (ou cuves) et d'engins de manutention (ponts roulants). La description détaillée d'une unité de traitement de surface sera abordée dans le paragraphe 1. On retrouve des unités de traitement de surface dans beaucoup de domaines allant de l'électronique (production des circuits imprimés) à l'aéronautique en passant par l'automobile.

La **spécificité** des opérations de traitement de surface par rapport à d'autres opérations, est une forte contrainte temporelle sur les opérations. Le temps opératoire est borné entre une durée

minimum, assurant la qualité de la pièce, et une durée maximum, induisant la non-qualité de la pièce. Dans certains cas, l'opération pourra être arrêtée en fin de traitement (traitements électrolytiques) ; dans d'autres cas, il faudra retirer la pièce de son emplacement (ou cuve) pour arrêter le traitement. Le détail de la problématique du pilotage des lignes de traitement de surface sera développé dans le paragraphe 2.

La contrainte de bornage des durées opératoires se retrouve également dans l'agroalimentaire et les procédés faisant intervenir des chauffes des produits, imposant un temps d'opération maximal avant l'opération suivante. Cette contrainte rend beaucoup d'algorithmes et d'heuristiques d'ordonnancement non applicables directement à cette problématique. On retrouve les premiers travaux dans ce domaine avec [Phillips L.-W. *et al.*, 1976], abordant l'automatisation d'une ligne de production de circuits imprimés. Un aperçu des méthodes de résolution, développées depuis, sera présenté dans ce chapitre.

1. Description des unités de traitement de surface

Une unité de traitement de surface est constituée d'un ensemble de cuves de traitement placées en ligne. Ces cuves sont desservies par un robot de manutention ('*pont roulant*' ou '*hoist*') situé au-dessus d'elles. Il a pour mission de déplacer les porteurs de cuve en cuve.

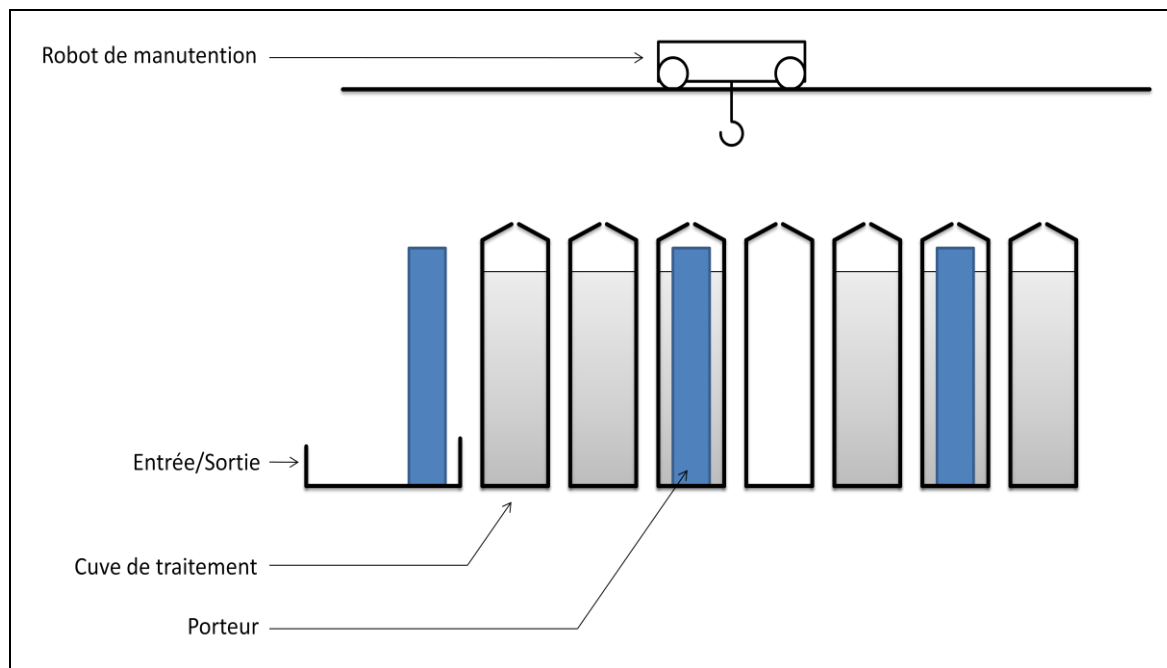


Figure 18 : Exemple de ligne de traitement de surface

1.1. Les porteurs

Une installation de traitement de surface est souvent une installation qui traite un grand nombre de pièces, avec régulièrement les mêmes gammes de fabrication. Nous verrons par la suite que la ressource critique de ces installations est souvent le robot de manutention. Afin de réduire le

nombre de transports, les pièces sont regroupées sur des porteurs. Ceux-ci ont l'avantage d'être adaptés au système d'accroche du robot.

On peut distinguer deux classes de porteur :

- Les **paniers** (ou tonneaux) dans lesquels les pièces sont mises en vrac.
- Les **cadres** auxquels les pièces sont suspendues.

Le choix d'un porteur et son chargement sont conditionnés par les pièces à transporteurs et par la gamme de traitement. Par exemple, dans le cas de pièces subissant des opérations manuelles de contrôle, toutes les faces doivent être accessibles. Dans ce cas, les pièces pourront être regroupées ou dégroupées en cours de gamme.

D'un point de vue pratique, l'ordonnancement des lignes de traitement de surface est conduit par les porteurs eux-mêmes. Ce sont eux qui sont conditionnés par une séquence d'opérations à réaliser.

Dans la suite de ces travaux, et en accord avec le langage métier de l'industriel aéronautique dans lequel ces travaux ont été effectués, nous appellerons indifféremment les porteurs '*porteur*' ou '*barre*'.

1.2. Les cuves

Les cuves sont les positions physiques dans lesquelles seront posés les porteurs. Leur mode de gestion et les contraintes temporelles qui leur sont liés dépendent du produit qu'elles contiennent. On distingue les classes suivantes :

- **Postes d'entrée sortie.** Ces postes ne contenant aucun produit sont les stocks d'entrée et de sortie de l'installation, ils sont l'interface entre l'unité de traitement de surface et l'extérieur. Ils sont habituellement alimentés et vidés manuellement par des ponts manuels. Les porteurs arrivent du poste d'accrochage des pièces et repartent vers le poste de décrochage des pièces. Ces postes peuvent être typés *entrée* ou *sortie*, ou non typés.
- **Postes de traitement.** Ces cuves sont les positions dans lesquelles les pièces vont subir une modification, une neutralisation, un contrôle... On peut distinguer :
 - **Les cuves de traitement.** Ces cuves contiennent les produits actifs qui vont nettoyer les pièces, modifier leur structure ou les préparer pour un contrôle. Dans le cas de cuves non activables, le traitement commencera dès l'immersion des pièces et s'arrêtera au moment de la sortie des pièces. Dans d'autres cas (traitements électrolytiques), le début et la fin de l'opération seront déclenchés par la mise en route du courant de l'électrolyse et son arrêt. Les tolérances sur les durées opératoires dans ces cuves sont souvent faibles.
 - **Les cuves de rinçage.** Ces cuves contiennent un produit de neutralisation d'un traitement. Les gammes opératoires y font passer les pièces après un traitement. Les produits qu'elles contiennent permettent d'avoir des tolérances sur les durées opératoires relativement grandes.

- **Les cuves de contrôle.** Ces cuves ne contiennent pas de produit. Elles permettent aux opérateurs d'effectuer un contrôle sur les pièces en milieu de gamme de traitement. On peut y effectuer des opérations de contrôles d'épaisseur, des contrôles de ressuage... Les durées opératoires pour un porteur dans ces cuves peuvent être variables en fonction de la disponibilité de l'opérateur au moment du contrôle, du nombre de pièces à contrôler.
- **Les postes d'attentes.** Ces positions sont identiques aux positions d'entrées et de sorties. Elles ne contiennent aucun produit et permettent de déposer des porteurs. Toutes les installations de traitement de surface ne disposent pas de position d'attente. Elles permettent de mettre en attente un palonnier. Le passage des porteurs par ces positions n'est souvent pas prévu dans la gamme. C'est l'ordonnanceur qui prend la décision du passage sur une position d'attente, afin de résoudre le problème d'affectation. Le passage par ces positions n'est pas toujours acceptable. Par exemple, il n'est pas possible d'y passer entre une opération de traitement chimique et une opération de neutralisation, car le produit précédent continuerait son activité sur les pièces pendant le passage au poste d'attente. Les gammes de traitement n'étant pas toujours linéaires et faisant parfois des retours en arrière, ces positions permettent d'éviter des blocages. La figure 19 montre un exemple de blocage : le *porteur 1* doit aller dans la *cuve B*, le *porteur 2* dans la *cuve E* et le *porteur 3* dans la *cuve A*. Dans ce cas, seul le passage d'un des porteurs par le poste d'attente C permet de débloquent le système. Un ordonnancement prédictif aurait pu éviter cette situation. De son côté, un ordonnancement réactif ne peut prévoir, à l'avance, ce blocage. Il doit y pallier en cours d'exécution.

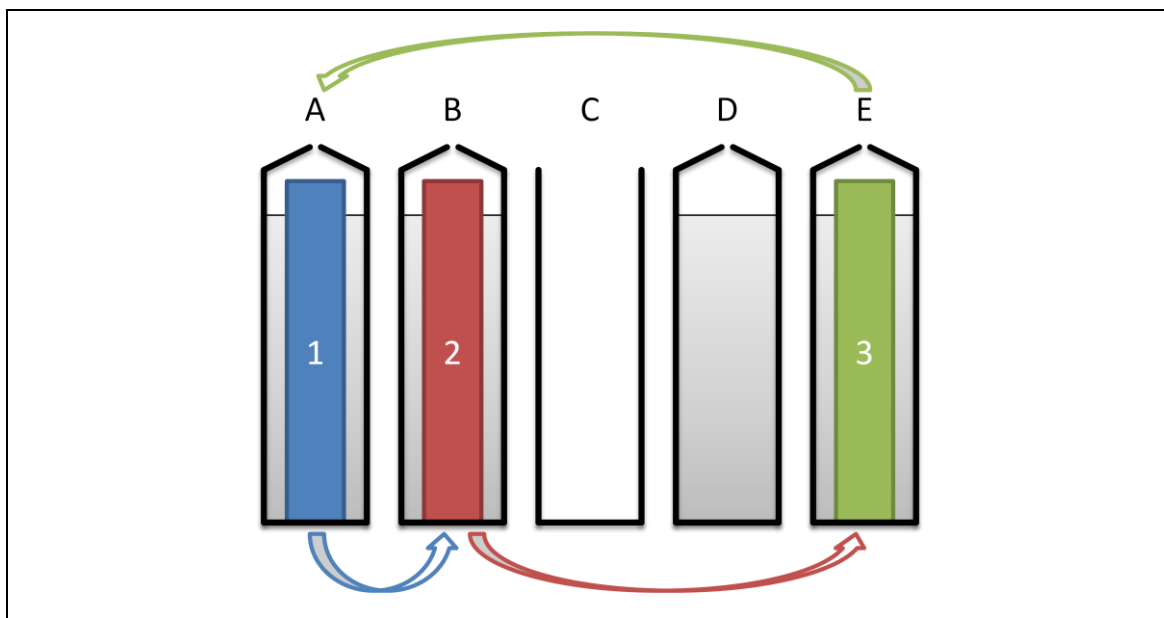


Figure 19 : Exemple de cas de blocage circulaire

Chacune des positions ayant des contraintes spécifiques de gestion, le problème d'ordonnancement d'une unité de traitement de surface dépend fortement du type de positions traitées. Certaines installations industrielles ne comportent pas de poste d'attente, ou de poste de contrôle manuel, ce qui peut changer la méthode de résolution.

Les différentes cuves sont (presque toujours) de capacité unitaire, c'est-à-dire qu'elles ne peuvent recevoir qu'un seul porteur à la fois. Afin de limiter les risques de ressource goulot, certaines cuves peuvent être dupliquées et réparties ou regroupées dans l'installation. On peut trouver deux cas d'application : les cuves multi-emplacements ou la duplication de cuves (voir figure 20). Le premier cas permet de traiter le problème en considérant les temps de transport identiques pour aller vers l'un ou l'autre des emplacements d'une cuve. Dans ce cas, les machines sont uniques, à capacité non-unitaire. Dans le second cas, l'approximation du temps de transport constant n'est plus applicable. Notons que la présence de cuves dupliquées pose un problème d'affectation multi ressource avec flexibilité, étudié dans la littérature, en autres dans [Dauzère-Pérès S. *et al.*, 1998] et [Lemoine D. *et al.*, 2010].

Un exemple de détermination du nombre optimal de cuves dupliquées dans une installation est présenté dans [Ng W.-C., 1995]. L'auteur se base sur un modèle mathématique du problème pour déterminer les cuves goulots, et leur nombre minimal de duplication. Le problème est résolu de façon successive en utilisant l'algorithme *branch and bound* proposé dans [Shapiro G.-W. *et al.*, 1988] jusqu'à trouver la meilleure configuration.

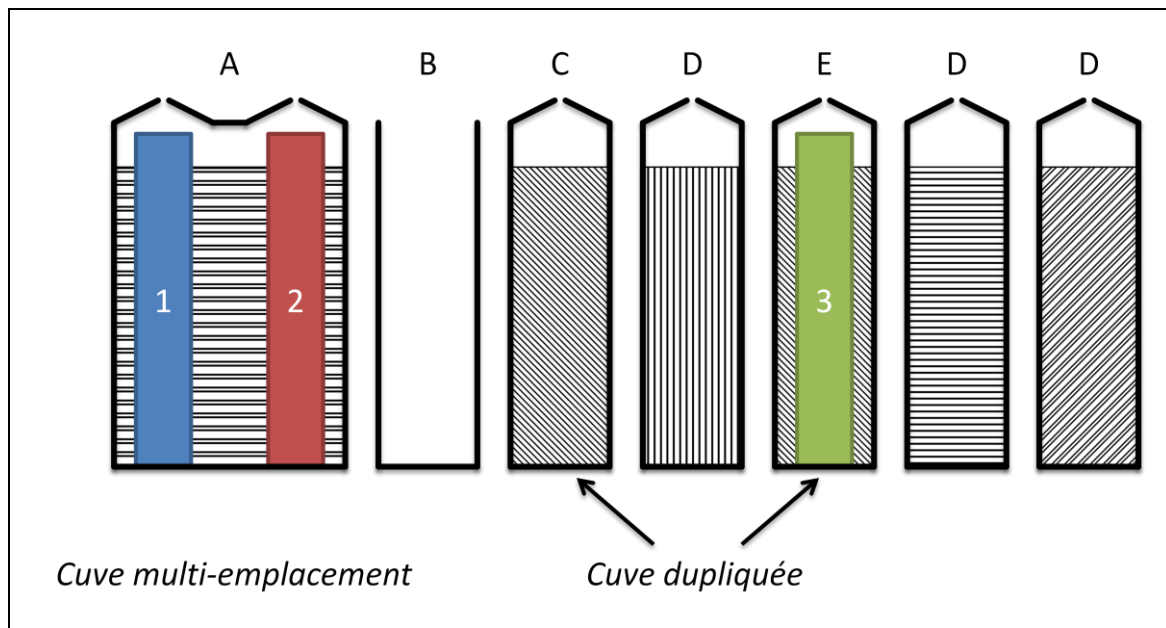


Figure 20 : Cuve multi emplacements ou cuves dupliquées

1.3. Les transporteurs (ponts roulants)

Les transporteurs (ou ponts roulants) sont les parties mobiles commandées de l'installation. Ils se déplacent sur un chemin de roulement afin de desservir différentes positions. On peut distinguer deux modes de pilotage des ponts :

- **les ponts manuels** utilisés pour apporter les porteurs dans les postes d'entrée. Ces ponts sont pilotés directement par les opérateurs.
- **les ponts automatiques** sont contrôlés par la supervision et reçoivent leurs missions du système d'ordonnancement.

Les missions d'un pont peuvent se décomposer en plusieurs phases élémentaires (figure 21 et figure 22) :

- **Phase 1** : déplacement à vide. Ce temps dépend de la distance de déplacement. Dans le cas où le pont est déjà à la bonne position, ce temps est nul.
- **Phase 2** : descente à vide.
- **Phase 3** : accrochage du porteur.
- **Phase 4** : montée en charge. La durée de cette phase dépend du bain et du type de pièces (et d'accroche).
- **Phase 5** : égouttage. La présence et la durée de cette phase dépendent du bain d'origine. Elle a pour but d'éviter la pollution prématurée des bains de rinçage par les produits chimiques restant sur les pièces.
- **Phase 6** : déplacement en charge. La durée de cette phase dépend à la fois de la distance parcourue et du type de porteur. En fonction des pièces accrochées et du système d'accroche, plusieurs vitesses peuvent être envisagées.
- **Phase 7** : descente en charge. Tout comme la phase 4, la durée de cette phase dépend de la position et du type de pièces et d'accroche.
- **Phase 8** : décrochage.
- **Phase 9** : montée à vide.

Dans le cas d'un traitement court, imposant de conserver le pont sur place, deux missions peuvent s'enchaîner sans remontée du pont. On trouve alors des missions de type *phase 1 à phase 7* suivie d'une mission de type *phase 4 à phase 9*.

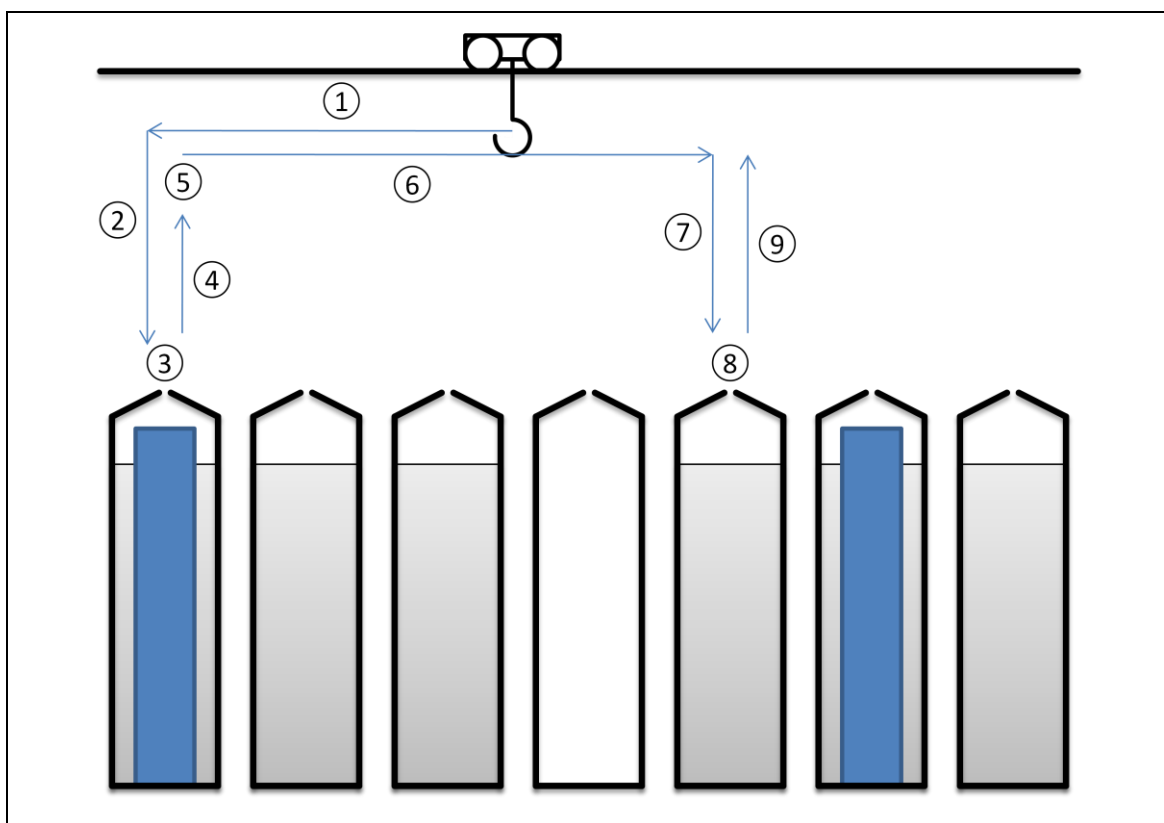


Figure 21 : Décomposition d'une mission de pont

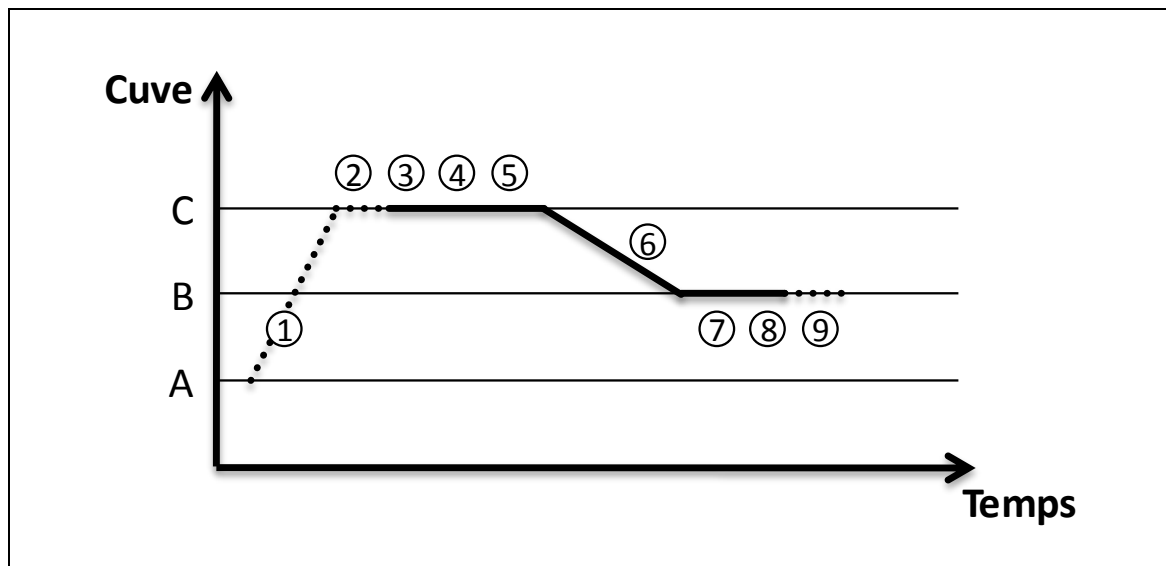


Figure 22 : Représentation d'un cycle robot sur un line-graph

La plupart des robots de manutention sont de **capacité unitaire**. Certains travaux ont abordé le cas de robots de manutention à capacité non unitaire. Drobouchevitch et ses co-auteurs, dans [Drobouchevitch I.-G. *et al.*, 2006], abordent le cas d'une cellule robotisée mono-robot **double-emplacement**. Ces travaux montrent l'augmentation de la complexité de l'ordonnancement d'un tel problème, mais souligne le gain de temps à pouvoir prendre une pièce traitée et déposer une pièce nouvelle dans un même cycle. Le cas traité ne prend en compte qu'un seul type de pièce, et cherche à trouver un ordonnancement cyclique. Dans une installation de traitement de surface, les enchainements rapides, imposés entre les traitements, empêchent le maintien d'un porteur sur le pont pendant une durée longue. On ne trouve pas, dans la littérature, de travaux d'ordonnancement des lignes de traitement de surface munies de robots multi-emplacements.

Dans le cas où les temps de transport et les temps de traitement sont du même ordre de grandeur, la ressource critique est généralement le pont. Cette criticité est renforcée par le progrès des traitements qui sont de plus en plus efficaces, donc de plus en plus courts. L'augmentation de la capacité de transport d'une installation se solde souvent par l'installation de plusieurs robots de manutention sur le même chemin de roulement. On parle alors de problème **multi-robots**. L'ordonnanceur doit alors gérer les conflits entre les robots, ce qui rend la résolution du problème plus complexe. Les travaux sur l'ordonnancement multi-pont se focalisent sur les exemples plus particuliers. Dans [Leung J.-M.-Y. *et al.*, 2006], les auteurs ont proposé un algorithme polynomial pour la détermination du nombre optimal de transporteurs. On peut remarquer que de nombreuses études portant sur le routage d'AGVs (*Automated Guided Vehicles*) sont applicables dans ce cas particulier. Afin de résoudre ce problème, les auteurs se sont placés dans le cas d'un ordonnancement cyclique d'une installation de traitement de surface avec des temps de traitement fixes (pas de tolérance). Ces contraintes, souvent réalistes dans le cas de fabrication de circuits imprimés en grande série, dégradent les performances dans le cas où des tolérances peuvent être appliquées aux durées de traitement.

Enfin, certaines installations de traitement de surface utilisent des porteurs de plusieurs types qui imposent au pont une *vitesse maximale en fonction des pièces transportées* pour éviter un

décrochage de celles-ci. Le temps de déplacement à vide (1) est lié à la distance tandis que le temps de déplacement en charge (6) est lié à la distance et au type de produit transporté.

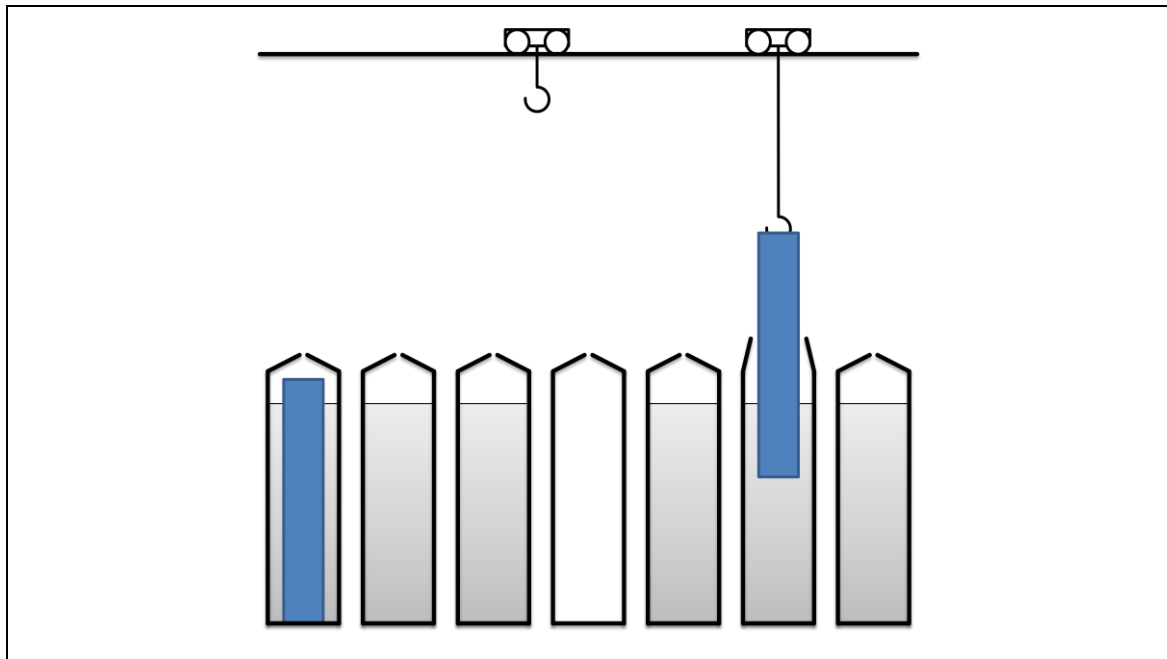


Figure 23 : Ligne de traitement de surface multi-robot

1.4. Les navettes

Une **ligne de traitement de surface** est constituée des différents postes desservis par un ou plusieurs ponts. Une **installation de traitement de surface** peut être constituée de plusieurs lignes en parallèles. Ces lignes peuvent être indépendantes, constituée chacune de postes d'entrée/sortie et de cuves de traitement, parfois connectées entre elles par des navettes. Les **navettes** sont des postes mobiles permettant de transporter des porteurs d'une ligne à l'autre. Les navettes pourront aussi servir de *poste d'entrée/sortie* dans le cas où les postes de préparation sont positionnés sur une ligne et les postes de traitement sur une autre. La figure 24 représente une unité constituée de deux lignes en parallèle.

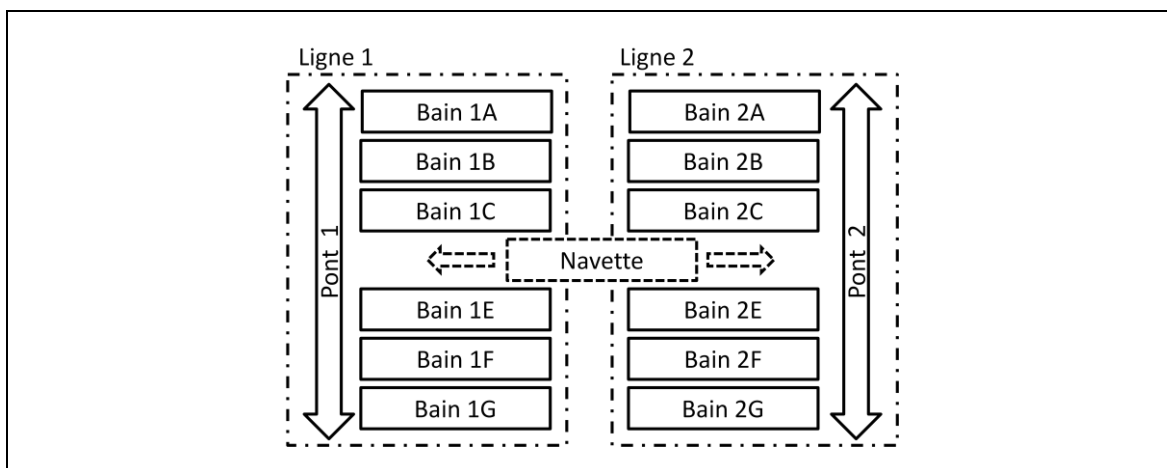


Figure 24 : Installation de traitement de surface avec navette

Dans [Rossé-Bloch C., 1999], l'auteur, reprenant la classification de [Manier M.-A., 1994], cite trois configurations types d'installations de traitement de surface : *en H*, *en O*, *en U*. Ces trois types sont représentés dans la figure 25.

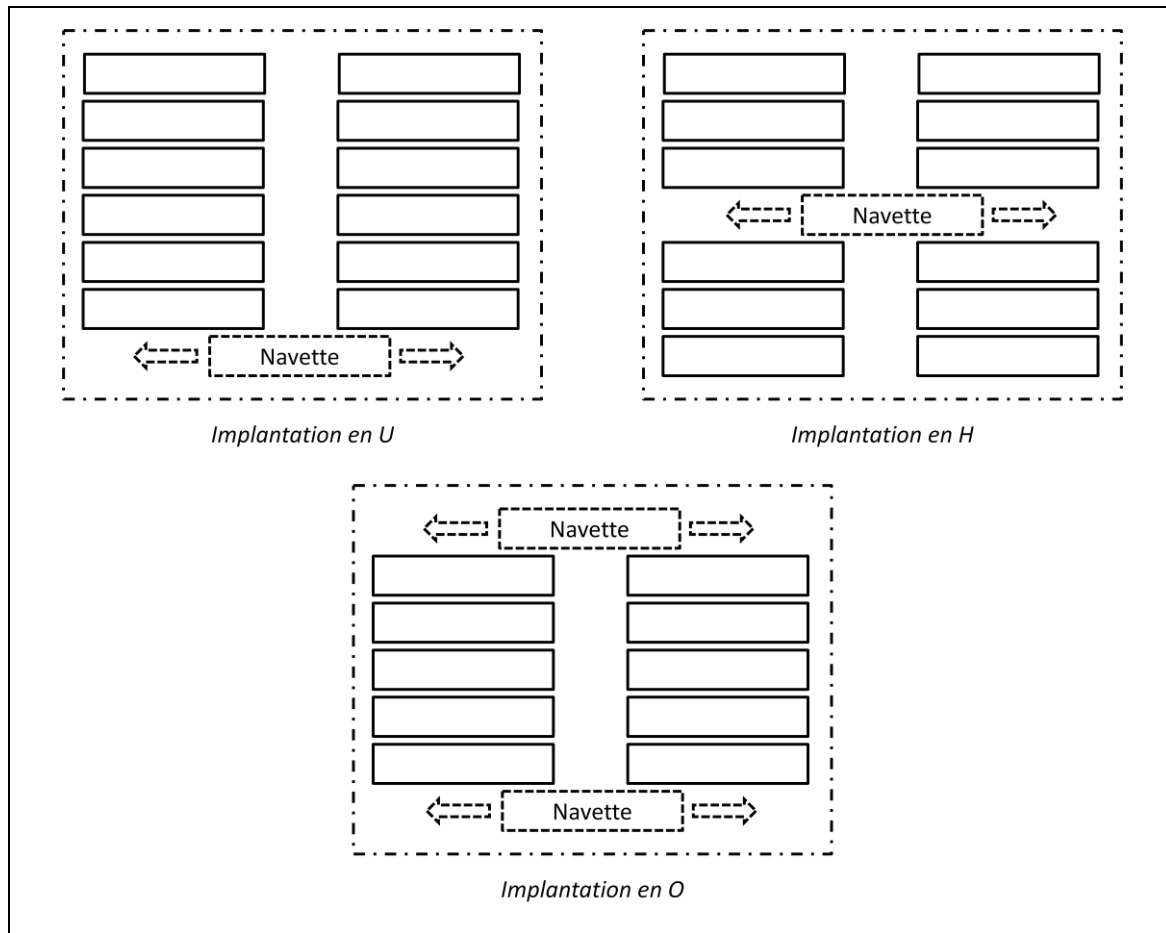


Figure 25 : Implantation type d'une unité de traitement de surface

Nous verrons par la suite que la présence de navettes est pénalisante pour les performances d'une unité. En effet, une navette introduit de fortes contraintes de synchronisation entre les différentes lignes, devenant alors les *machines goulot*, limitant la productivité.

2. Problématique du Hoist Scheduling Problem

2.1. Définition d'une gamme de traitement

Une gamme de traitement de surface est une succession d'opérations de trempe et d'opérations de déplacement. Elle commence toujours par l'arrivée d'un produit sur un poste de chargement et se termine sur un poste de déchargement (les postes de chargement/déchargement peuvent être indifférenciés). Chaque durée opératoire se trouve dans une fenêtre temporelle correspondant à la durée minimum et maximum de traitement. De plus, dans le cas de traitements non critiques (rinçage), il est possible, mais non obligatoire, de passer par un poste d'attente.

| Opération | Durée min | Durée max |
|-----------------------------|---------------|---------------------|
| <i>Chargement</i> | <i>0</i> | <i>infinie</i> |
| <i>Attente (facultatif)</i> | <i>0</i> | <i>infinie</i> |
| <i>Traitement A</i> | <i>50 min</i> | <i>52 min</i> |
| <i>Rinçage A</i> | <i>5 min</i> | <i>30 min</i> |
| <i>Attente (facultatif)</i> | <i>0</i> | <i>infinie</i> |
| <i>Traitement B</i> | <i>5 min</i> | <i>5 min 20 sec</i> |
| <i>Rinçage B1</i> | <i>10 min</i> | <i>15 min</i> |
| <i>Rinçage B2</i> | <i>10 min</i> | <i>15 min</i> |
| <i>Attente (facultatif)</i> | <i>0</i> | <i>infinie</i> |
| <i>Déchargement</i> | <i>0</i> | <i>infinie</i> |

Tableau 1 : Exemple de gamme de traitement

On peut alors noter les caractéristiques suivantes qu'on retrouve dans beaucoup de problèmes de traitement de surface :

- Les gammes de traitement sont composées de groupes (ici le groupe A et le groupe B).
- Chaque groupe commence par une opération de traitement critique.
- Il peut y avoir zéro, un ou deux rinçages par groupe.
- Les opérations au sein d'un groupe doivent s'enchaîner, sans passage par une attente.
- Le passage par une position d'attente est possible entre les groupes d'opérations.

Notons ici qu'une opération de traitement est affectée à un groupe de ressources capable de réaliser le traitement. Dans le cas de problèmes multi-bains (cuves dupliquées), l'affectation à telle ou telle cuve n'est pas (en général) fixée a priori.

2.2. Notations

On distingue ici les données de configuration du système et de la production à réaliser (données du problème) et les résultats de l'ordonnancement.

2.2.1. Données du problème

- Poste
 - p Identifiant du poste (numéro d'emplacement)
 - Np Nombre de postes
 - TS_p Type de traitement du poste p ($p \in \llbracket 1; Np \rrbracket$)
- Robot de manutention
 - $DeplV_{p1,p2}$ Temps de déplacement à vide du poste $p1$ au poste $p2$ (phase 1)
 - $TpsMV_p$ Temps de montée à vide en p (phase 2)
 - $TpsDV_p$ Temps de descente à vide en p (phase 9)
- Gamme
 - g Identifiant de la gamme
 - o Identifiant de l'opération ($o \in \llbracket 1; No_g \rrbracket$)
 - No_g Nombre d'opérations dans la gamme
 - $TS_{g,o}$ Type de traitement
 - $tMin_{g,o}$ Durée minimum de traitement
 - $tMax_{g,o}$ Durée maximum de traitement
 - $Egout_{g,o}$ Durée d'égouttage (phase 5)
 - $Crit_{g,o}$ Criticité de l'opération
- Palonnier
 - b Identifiant du palonnier
 - g_b Numéro de gamme du palonnier
 - $DeplC_{b,p1,p2}$ Temps de déplacement en charge du poste $p1$ au poste $p2$ (phase 6)
 - $TpsMC_{b,p}$ Temps de montée en charge en p (phase 4)
 - $TpsDC_{b,p}$ Temps de descente en charge en p (phase 7)
 - $Acc_{b,p}$ Temps d'accrochage (phase 3)
 - $Dec_{b,p}$ Temps de décrochage (phase 8)

2.2.2. Résultat de l'ordonnancement

- Palonnier
 - $p_{b,o}$ Identifiant du poste de l'opération
 - g_b Gamme associée
 - $tDeb_{b,o}$ Date de début de l'opération o
 - $tFin_{b,o}$ Date de fin de l'opération o
 - $m_{b,o}$ Identifiant de la mission de sortie
 - $m_{b,o-1}$ Identifiant de la mission d'entrée
- Robot de manutention
 - m Identifiant de la mission
 - Nm Nombre de missions
 - $pDeb_m$ Poste de prise
 - $pFin_m$ Poste de dépose

- $tDeb_m$ Date de début de la mission
- $tFin_m$ Date de fin de la mission

2.3. Contraintes d'ordonnancement

Dans [Baptiste P. *et al.*, 1992] (problème général), [Baptiste P. *et al.*, 1996] (problème cyclique) et [Manier M.-A. *et al.*, 2000] (problème cyclique), les auteurs ont proposé un modèle de contraintes du HSP pour la résolution du problème par PLC (*Constraints Logic Programming*). Dans [Hindi K.-S. *et al.*, 2004], les auteurs proposent une heuristique de construction pas à pas d'un planning, avec contrôle des contraintes à chaque étape (*Constraint Satisfaction Problem*). Les modèles présentés permettent d'une part, de résoudre le problème abordé dans chaque article, et d'autre part, de vérifier les solutions issues d'autres méthodes de résolution. Les contraintes sont reprises dans le tableau 2.

| Contrainte | Modèle mathématique associé |
|---|---|
| Respect des types de traitement selon la gamme de la barre. | $\forall(b, o)$ $TS_{p_{b,o}} = TS_{g_{b,o}}$ |
| Respect des durées de traitement dans les fenêtres temporelles. | $\forall(b, o)$ $tFin_{b,o} - tDeb_{b,o} \in [tMin_{g_{b,o}} ; tMax_{g_{b,o}}]$ |
| Dégagement en attente possible pour un traitement non critique, enchainement obligatoire pour un traitement critique. | $\forall(b, o)$ $TS_{pFin_{m_{b,o}}} \begin{cases} TS_{g_{b,o}} & \text{si } Crit_{g_{b,o}} = \text{Vrai} \\ \text{Attente} & \\ \text{ou} & \\ TS_{g_{b,o}} & \end{cases} \quad \text{sinon}$ |
| Deux produits ne peuvent se trouver au même moment au même endroit. | $\forall(b1, o1, b2, o2) / b1 \neq b2 \wedge p_{b1,o} = p_{b2,o}$ $[tDeb_{b1,o}; tFin_{b1,o}] \cap [tDeb_{b2,o}; tFin_{b2,o}] = \emptyset$ |
| Un déplacement du pont ne doit pas rentrer en conflit avec un autre déplacement. | $\forall(m1, m2) / m1 \neq m2$ $[tDeb_{m1}; tFin_{m1}] \cap [tDeb_{m2}; tFin_{m2}] = \emptyset$ |
| Il faut suffisamment de temps entre deux missions de pont pour effectuer un déplacement à vide. | $\forall(m1, m2) / tDeb_{m2} > tFin_{m1}$ $tDeb_{m2} - tFin_{m1} > DeplV_{pFin_{m1}, pDeb_{m2}}$ |

Tableau 2 : Tableau des contraintes d'ordonnancement

2.4. Fonction objectif

Le but d'une unité de traitement de surface est de produire des pièces répondant aux contraintes du procédé. Afin d'obtenir ce résultat, plusieurs fonctions objectifs peuvent être envisagées. L'optimal de la '*fonction objectif*' est, en général, issu d'une exploration du champ des solutions répondant aux contraintes énoncées dans le paragraphe 2.3. Dans certains cas, l'exploration n'est pas contrainte. La solution doit être consolidée afin de répondre aux contraintes (c'est le cas de certaines applications des algorithmes génétiques).

La première fonction objectif, qui est la plus répandue dans la littérature sur l'ordonnancement (dont l'ordonnancement des lignes de traitement de surface) est la minimisation de la date de sortie du dernier produit, souvent appelé *makespan* et noté C_{max} dans la littérature :

$$C_{max} = \min(tFin_{Nm})$$

Cette fonction revient à maximiser la cadence moyenne de sortie des palonniers, sans s'attarder sur ce qui se passe localement dans l'installation. Corine Subaï et ses co-auteurs dans [Subaï C. *et al.*, 2003] puis dans [Subaï C. *et al.*, 2006] rappellent que les opérations de traitement de surface ne sont pas des opérations de fabrication anodines. Elles mettent en jeu des produits chimiques qui sont altérés avec le temps. Cela soulève deux problèmes :

- La concentration est l'activité des produits évoluant au fur et à mesure des différents traitements, **la durée de traitement $tMin_{g,o}$ est a priori variable**. Ce paramètre n'est pas pris en compte dans la plupart des ordonnancements. Ceci signifie que la durée utilisée pour ordonnancer n'est pas la durée optimale, ce qui induit une présence dans les bains, supérieure au minimum chimiquement nécessaire.
- La présence de tolérance dans le problème d'ordonnancement introduit, de fait, la possibilité de rester plus longtemps dans une cuve de traitement que la durée minimum : $tMin_{g,o} < tFin_{b,o} - tDeb_{b,o} < tMax_{g,o}$

Ces deux phénomènes conduisent à un vieillissement prématuré de la solution chimique, donc à une plus grande pollution. Les auteurs proposent de chercher à minimiser les temps inutiles de traitement. Ils démontrent qu'en gardant le même objectif de productivité, il reste une certaine latitude sur les dates de début et de fin d'opérations, donc sur le vieillissement prématuré ou non des solutions. L'objectif devient alors la minimisation des temps supplémentaires passés dans une cuve :

$$\min \left(\sum_{b,o} tFin_{b,o} - tDeb_{b,o} - tMin_{g,b,o} \right)$$

Les cuves n'étant pas toutes constituées des produits ayant le même impact écologique, il est possible de pondérer chacun des termes avec un facteur de pollution.

La première étape vers le pilotage propre des lignes de traitement de surface consiste à conserver la productivité optimale (première fonction objectif) puis à chercher la réduction des temps inutiles de

traitement. La deuxième étape, axée vers le développement durable, consiste à faire une optimisation multi objectifs sur ces deux fonctions. D'un point de vue global, il faudrait ajouter dans la balance économique le coût représenté par le recyclage des solutions (non négligeables). La deuxième *fonction objectif* n'est donc pas forcément moins rentable pour l'entreprise.

2.5. Intégration du HSP dans les problèmes d'ordonnancement

Le problème de l'ordonnancement des lignes de traitement de surface est un problème d'ordonnancement d'atelier avec quelques contraintes particulières que l'on peut retrouver dans d'autres problèmes. Afin de permettre des échanges entre les spécialistes de l'ordonnancement d'ateliers et les chercheurs travaillant sur les problèmes de traitement de surface, il est utile de replacer le HSP au sein des problèmes d'ordonnancement.

Le jargon industriel utilisé pour caractériser un problème réel de traitement de surface n'étant pas toujours bien défini, l'adéquation entre le *parler métier* et la *terminologie scientifique* n'est pas toujours exacte. Ainsi, certaines contraintes industrielles telles que '*dans un bain critique il faut respecter la tolérance ; dans il bain non critique, il est bien de la respecter*' peut devenir pour les scientifiques un vrai casse-tête de modélisation. La précision de telles affirmations est scientifiquement nécessaire, mais peut devenir une contrainte supplémentaire. Le langage métier est à replacer dans la culture de l'entreprise pour comprendre efficacement les objectifs recherchés, et ne pas perdre d'information par une modélisation trop stricte de la contrainte.

Christelle Rossé-Bloch, dans [Rossé-Bloch C., 1999], a cherché à resituer le problème de l'ordonnancement des lignes de traitement de surface par rapport aux problèmes généraux d'ordonnancement. Nous reprendrons ici les grandes lignes de ces travaux, et vous invitons à vous y référer pour plus de détails. On y retrouve un dictionnaire de terminologie repris dans le tableau 3.

| | Terminologie classique | Vocabulaire du HSP |
|---------|--|--|
| Données | travaux / jobs | produit – pièce / porteur |
| | tâche / opération élémentaire | opération de trempe / opération de transport |
| | ressource : - disjonctive - cumulative | - robot mono préhenseur - cuve mono-bac - robots, cuves multi-bacs, porteurs |

| | Terminologie classique | Vocabulaire du HSP |
|------------------------------|---------------------------|---|
| Contraintes | potentielles : | succession sans attente et sans préemption entre : |
| | - antériorité | - opérations de trempe - opération de trempe et opération de transport |
| | - localisation temporelle | - durée entre deux trempes - durée entre deux transports |
| | disjonctives | capacité unitaire : - chaque robot (mono-préhenseur) - chaque cuve mono-bac |
| | cumulatives | capacité non unitaire : - un seul robot (double préhenseur) - plusieurs robots - chaque cuve multi bacs (postes de chargement par exemple) - porteurs |
| Objectifs et critères | | maximiser la productivité (C_{max} , débit) |

Tableau 3 : Terminologie du HSP selon [Rossé-Bloch C., 1999]

Les pièces sont regroupées par gamme de traitement sur un porteur. On peut donc, soit considérer l'ordonnancement des pièces en prenant en compte leur placement sur les porteurs, soit n'aborder que l'ordonnancement des porteurs, caractérisé par la gamme de fabrication commune à toutes les pièces portées. Les problèmes de placement et d'ordonnancement étant des problèmes complexes, ils ont souvent été traités séparément, le résultat du premier contraignant la résolution du second.

On trouve deux types de ressources dans ce problème : les postes et les transporteurs. Les postes subissent les déplacements réalisés par les transporteurs, on peut donc restreindre le problème à la résolution de l'ordonnancement des transports. Les opérations de trempe seront déduites des opérations de transport par complémentarité. On peut alors réaliser le parallèle avec l'ordonnancement des **transporteurs** ou l'ordonnancement d'**une machine à temps de production nul et temps de setup variable**. La matrice des temps de setup serait alors la matrice des temps de déplacement du transporteur. Le problème peut alors se rapprocher du problème fondateur sur les machines à temps de setup variable proposé dans [White C.-H. *et al.*, 1977]. Dans l'approche de résolution proposée, l'objectif est la minimisation des temps de réglage, ce qui correspond dans un

problème de HSP à une minimisation des temps de déplacement. Cette approche serait applicable si on pouvait modéliser la contrainte des temps de traitement dans le problème d’ordonnancement à une machine. Ces contraintes ne sont actuellement pas prises en compte dans la résolution des problèmes de machines à temps de setup variable.

3. Classification des problèmes

Les travaux sur le problème d’ordonnancement des lignes de traitement de surface sont très nombreux depuis les années 1970. Il existe autant de problèmes que d’installations de traitement. En effet, chaque installation a des contraintes spécifiques. C’est ainsi qu’un industriel aura des cuves manuelles avec des durées de passage non prévisibles, un autre aura plusieurs ponts avec une gestion spéciale des conflits, un autre encore aura une production cyclique, et un chercheur aura un problème académique complexe.

Manier et Bloch ont proposé dans [Bloch C. *et al.*, 1999] puis dans [Manier M.-A. *et al.*, 2003] une extension des classifications de problèmes d’ordonnancement de [Rinnoy Kan A.-H.-G., 1976], [Graham R.-L. *et al.*, 1979] et [Blazewicz J. *et al.*, 1993]. Cette extension est appliquée aux problèmes d’ordonnancement des lignes de traitement de surface. Les classifications servant de référence ne prennent pas en compte les fenêtres temporelles sur les durées de traitement, les déplacements à vide, la gestion de plusieurs ponts, la capacité des cuves, la gestion des porteurs vides... autant de paramètres qui sont spécifiques au traitement de surface. Des extensions des notations initiales adaptées aux ateliers flexibles avaient été développées mais sans prendre en compte toutes les contraintes du HSP.

La classification proposée se base sur 4 champs :

$$\alpha \mid \beta \mid \delta \mid \gamma$$

Le **champ α** est une extension du champ α des classifications des problèmes d’ordonnancement citées ci-dessus. On lui définit quatre nouvelles valeurs de la forme $XHSP$. Il décrit le type de HSP pris en compte. Le terme *HSP* implique l’existence de limites inférieures et supérieures pour les durées des traitements, l’interdiction de la préemption des ressources cuve ou pont, et impose des contraintes d’anticollision des ponts et la capacité unitaire des ponts. Ces contraintes interdisent les cuves manuelles, ou du moins, imposent qu’on leur fixe des durées de traitement. On trouve quatre classes de HSP dans la littérature et l’industrie :

- Le **PHSP** (*Predictive Hoist Scheduling Problem*) consiste à établir un ordonnancement des opérations de déplacement et de trempe avant l’exécution de la production. Il a l’avantage de permettre une optimisation a priori, mais ne prend pas en compte les aléas ou l’arrivée de nouvelles pièces. Il est réalisé à horizon fini pour une production donnée.
- Le **CHSP** (*Cyclic Hoist Scheduling Problem*) cherche à ordonnancer la production de pièces dont l’arrivée est cyclique. L’objectif est de chercher un motif d’ordonnancement qui sera répété dans le temps. Ce problème s’applique aux

installations ayant peu de gammes différentes (souvent une ou deux), et dont la production est répétitive. Cette classe de problème est la plus abordée dans la littérature.

- Le **DHSP** (*Dynamique Hoist Scheduling Problem*) est basé sur la définition d'un ordonnancement que l'on peut remettre en cause en cours de production afin de prendre en compte de nouveaux ordres de fabrications ou des aléas. Le caractère *en-ligne* de ces problèmes impose le choix d'heuristiques à convergence rapide.
- Le **RHSP** (*Reactive Hoist Scheduling Problem*) est basé sur un pilotage sans prévision. Nous verrons dans le chapitre suivant un exemple de RHSP, et les problèmes sous-jacents.

Le choix de l'une ou l'autre des méthodes de résolution à partir d'un problème donné provient des contraintes liées à l'environnement étudié :

| | | L'environnement est-il soumis à des perturbations ? | |
|-------------------------------|-----|---|------|
| | | Oui | Non |
| Le problème est-il cyclique ? | Oui | DHSP ou RHSP* | CHSP |
| | Non | DHSP ou RHSP* | PHSP |

* Ce choix sera fait en fonction de la fréquence d'occurrence des perturbations et du temps de calcul nécessaire au DHSP pour établir un nouveau planning.

Tableau 4 : Choix du champ α de [Manier M.-A. et al., 2003]

Le **champ β** concerne la description physique de l'installation du système. Il est de la forme :

$$\beta_1, \beta_2, \beta_3, \beta_4[, \beta_4 \dots] / \beta_5, \beta_6, \beta_7, \beta_8 / \beta_9$$

La description de ces champs est reprise dans le tableau 5. Dans le cas où un champ prend la valeur par défaut, on peut indiquer \emptyset ou ne pas le faire apparaître.

| | Forme | Description | Défaut |
|-----------|---|--|-------------|
| β_1 | $\beta_1 \in \mathbb{N}^*$ | Nombre de lignes considérées dans le problème. | 1 |
| β_2 | $\beta_2 \in \mathbb{N}$ | Nombre de mécanismes de transfert. | 0 |
| β_3 | $\beta_3 \in \{\emptyset, \text{synchro}\}$ | Nécessité de synchroniser les ponts et les transferts. | \emptyset |

| | Forme | Description | Défaut |
|-----------|---|--|-------------|
| β_4 | $\beta_4 = (\beta_{4,1}, \beta_{4,2}, \beta_{4,3})$ $\in (\mathbb{N}^*, \mathbb{N}^*, \mathbb{N}^*)$ | Décrit chaque ligne (un paramètre β_4 par ligne) : <ul style="list-style-type: none"> - $\beta_{4,1}$ nombre de transporteurs - $\beta_{4,2}$ nombre de bays (excepté chargement/déchargement) - $\beta_{4,3}$ capacité maximum des bays | 1 1 |
| β_5 | $\beta_5 \in \{\emptyset\} \cup \mathbb{N}^*$ | Nombre de porteurs. \emptyset indique qu'on ne gère pas les porteurs. | \emptyset |
| β_6 | $\beta_6 \in \{\emptyset, circ\}$ | Définit pour un CHSP la nécessité de recharger un transporteur avant de commencer un nouveau cycle. | \emptyset |
| β_7 | $\beta_7 \in \{\emptyset, ret\}$ | Définit la nécessité de déplacer un porteur de la station de déchargement vers la station de rechargement (dans le cas où ces stations sont différenciées). | \emptyset |
| β_8 | $\beta_8 \in \{\emptyset, empty\}$ | Définit le risque de manque de place pour les transporteurs. <i>empty</i> indique la possibilité de faire circuler des porteurs vides dans l'installation pour éviter ce manque de place. \emptyset indique qu'il n'y a pas de manque, ou que des places sont disponibles à côté de l'installation. | \emptyset |
| β_9 | $\beta_9 \in \{\emptyset, ass, diss\}$ | Décrit le fonctionnement des postes de chargement/déchargement : <ul style="list-style-type: none"> - <i>ass</i> les postes de chargement/déchargement sont gérés comme les autres postes physiques (capacité fixe) - <i>diss</i> les postes de chargement/déchargement sont dissociés des positions physiques - \emptyset le mode de gestion n'est pas important | \emptyset |

Tableau 5 : Description du champ β de [Manier M.-A. et al., 2003]

Le **champ γ** décrit l'environnement de production considéré. Il est de la forme :

$$\delta_1 / \delta_2, \delta_3, \delta_4, \delta_5$$

| | Forme | Description | Défaut |
|------------|--|---|-------------|
| δ_1 | $\delta_1 \in \{\emptyset\} \cup \mathbb{N}^*$ | Nombre de pièces ou de lots traités. \emptyset signifie qu'une infinité de pièces peut être traitée d'affilée sur l'installation. | \emptyset |

| | Forme | Description | Défaut |
|------------|--|---|-------------|
| δ_2 | $\delta_2 \in \{\emptyset\} \cup \mathbb{N}^*$ | Nombre de gammes différentes. \emptyset signifie que tous les produits ont la même gamme. | \emptyset |
| δ_3 | $\delta_3 \in \mathbb{N}^*$ | Nombre d'opérations de la plus longue gamme. | |
| δ_4 | $\delta_4 \in \{\emptyset, clean\}$ | Spécifie si les porteurs doivent être nettoyés après déchargement. | \emptyset |
| δ_5 | $\delta_5 \in \{\emptyset, recr\}$ | Définit la contrainte de recirculation dans le cas de bains multifonctions. \emptyset signifie que tous les bains sont mono-fonction. | \emptyset |

Tableau 6 : Description du champ γ de [Manier M.-A. et al., 2003]

Le **champ γ** est le critère à optimiser. Son expression est libre et dépend de l'objectif fixé par le problème. Il peut, par exemple, être égal à C_{max} dans le cas où l'objectif est de minimiser le makespan.

[Manier M.-A. et al., 2003] et [Rossé-Bloch C., 1999] ont proposé d'appliquer cette classification à des nombreux travaux de recherche afin d'établir une typologie des méthodes de résolution trouvées dans la littérature. Cette typologie classe les différents travaux en fonction de leur paramètres α , β , δ et γ . Nous utiliserons cette classification pour présenter les unités de traitement de surface abordées dans cette thèse.

4. Approches de résolution

Beaucoup de méthodes d'ordonnancement ont été appliquées depuis les années 1970 pour résoudre le problème du pilotage des lignes de traitement de surface. Le choix de l'une ou l'autre des méthodes est souvent lié aux contraintes du problème et aux simplifications prises en compte. Nous présenterons dans cette partie quelques travaux en les classant par types d'ordonnements réalisés (champ α de la classification présentée dans la partie 3).

4.1. Méthodes hors ligne

4.1.1. Ordonnement cyclique

Le problème du CHSP (*cyclic hoist scheduling problem*) est le HSP le plus traité dans la littérature. C'est un cas d'ordonnement prédictif réalisé à horizon temporel infini. Le RHSP consiste à ordonner une production dont l'arrivée des pièces est cyclique, dont la qualité des bains est stable (afin d'avoir des durées de traitement constantes) et dont le système ne présente pas d'alea.

L'objectif est de trouver un motif reproductible périodiquement (voir un exemple figure 26) avec, dans la majorité des travaux, une recherche de durée de cycle minimale. Un alea sur l'arrivée des

pièces, sur un bain ou une modification des paramètres, conduit très souvent au déchargement complet et au rechargement de l'unité de traitement. Les phases transitoires (chargement/déchargement de l'unité) étant passagères, elles ne sont que très rarement étudiées.

[Levner E. *et al.*, 2010] réalise un état de l'art des différents problèmes d'ordonnancements cycliques. Les auteurs soulignent que, contrairement au problème job-shop mono-produit qui est NP-complet, le CHSP mono-produit est un problème de complexité polynomiale. Les auteurs de [Kats V. *et al.*, 2008], en établissant le lien entre le problème d'ordonnement cyclique et la recherche d'un plus court chemin dans un graphe orienté, ont démontré la complexité polynomiale du problème d'ordonnement cyclique mono-robot à temps de production borné. La complexité polynomiale n'étant pas liée directement au temps de calcul mais à son accroissement en fonction de la taille du problème, la résolution par exploration exhaustive des solutions n'est pas toujours possible. Le recours à des algorithmes d'exploration partielle est alors nécessaire, nous présenterons ci-dessous quelques travaux appliqués au CHSP utilisant des algorithmes d'exploration connus.

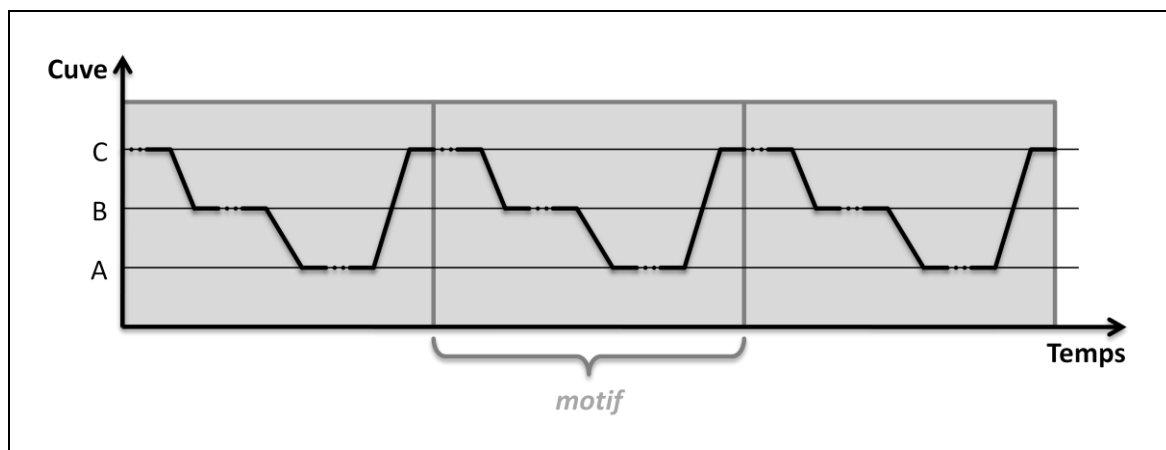


Figure 26 : Exemple d'ordonnement cyclique d'une unité à trois cuves

Les premiers travaux sur le HSP ont été réalisés sur un problème cyclique et présentés dans [Phillips L.-W. *et al.*, 1976]. La méthode proposée est basée sur un algorithme **branch and bound** sur un arbre binaire, mais l'espace des solutions explorées a été volontairement très réduit, ce qui empêche la convergence certaine vers la solution optimale. Ces travaux ont servi de base à la modélisation du HSP et ont inspiré beaucoup d'autres travaux sur le CHSP, notamment pour l'utilisation d'un algorithme *branch and bound*. Les travaux qui ont suivi ont permis d'explorer d'autres problèmes sans contraindre le champ de solution, ou en le contraignant de façon à ne pas détériorer la solution optimale.

Dans [Che A. *et al.*, 2005], les auteurs proposent un modèle polynomial de résolution du CHSP à un seul pont, avec prise en compte des fenêtres temporelles et des durées de transport. Dans [Che A. *et al.*, 2007]⁶, les auteurs ont ensuite proposé d'aborder la résolution du CHSP sans fixer *a priori* le nombre de cuves dupliquées. La méthode de résolution proposée consiste à lister des propriétés mathématiques de la solution finale (nombre maximal de pièces traitées simultanément, borne

⁶ CHSP | , , , (∅, 10, n) // | / recr | Cmax

minimale et maximale du temps de cycle, nombre maximal de cuves dupliquées) afin de déterminer un modèle mathématique contraint du problème, et donc de réduire le champ des solutions envisageables. L'espace des solutions est ensuite exploré à l'aide d'un algorithme **branch and bound**. Les variables de l'espace exploré sont, d'un côté, le nombre de duplications de cuves et, de l'autre côté, l'ordre des opérations de déplacement du transporteur. La faisabilité et les performances de chaque solution sont évaluées à l'aide du modèle mathématique précédemment établi.

[Shilong W. *et al.*, 2009] s'est penché sur le CHSP sans attente. La méthode utilisée repose sur l'hybridation d'un **algorithme génétique** avec une recherche locale basée sur le **recuit simulé**. L'objectif fixé est de trouver le meilleur makespan, donc la meilleure productivité. Chaque solution est représentée par un chromosome. Différentes solutions forment alors une population qui subira des *croisements* et des *mutations* afin de produire, par itérations successives, le meilleur individu. Le recuit simulé est ensuite utilisé pour optimiser la solution issue de l'*algorithme génétique* dans son voisinage local. Cette méthode appliquée à d'autres problèmes d'ordonnancement est applicable dans le cas d'un problème sans attente, mais devient inapplicable dans le cas d'un HSP avec attente. Cette contrainte, détériorant grandement la solution par rapport à des méthodes prenant en compte les tolérances, ne sera utile que pour les installations sans attente.

4.1.2. Ordonnancement prédictif

Rappelons ici que l'**ordonnancement prédictif** consiste à établir, avant le lancement de la production, un planning figé à horizon fini de l'ensemble des opérations de trempe et transport. Dans ce cadre, aucune remise en cause du planning n'est possible et un aléa stoppera la production. Cette classe de problème n'est applicable que dans un **environnement déterministe**, où l'ensemble de la production et les réactions du système physique seraient connus à l'avance. Enfin, le caractère **hors-ligne** de ces méthodes rend leur temps de calcul dissocié du temps de production, ce qui permet la recherche d'une **solution optimale**.

Bushan et Karimi ont abordé le problème de l'ordonnancement des lignes de traitement de surface à l'aide de deux approches : une approche basée sur la programmation linéaire en nombre entier et une approche basée sur les méta-heuristiques. Les deux travaux portent sur l'impression de circuits imprimés qui ont tous le même séquençement. Ce problème serait cyclique si les auteurs n'avaient pas pris en compte la détérioration de la qualité des bains, et donc la variabilité des temps de traitement. Il y a alors autant de gammes de traitement qu'il y a de pièces.

Dans [Bhushan S. *et al.*, 2003], les auteurs ont abordé le problème à l'aide d'outils de **programmation linéaire sous contrainte**. Le problème a été découpé en deux sous-problèmes : la première étape consiste à déterminer l'ordre des opérations de trempe et leur date de début tandis que la seconde étape ajoute la contrainte liée à l'unicité du robot, et donc à la suppression des conflits éventuels sur les opérations de transport. L'ajout de la contrainte d'unicité du robot dans la première étape rend le calcul entre 2 et 2000 fois plus lent, et ce facteur croît exponentiellement avec l'importance du problème. C'est ce constat qui a conduit les auteurs à découper le problème en deux étapes. La première étape n'étant pas résoluble en un temps acceptable, les auteurs ont proposé de réduire arbitrairement le champ des solutions au fur et à mesure de l'avancée du temps

CPU, puis en relaxant les contraintes estimées comme non-essentiels. Cette méthode assure une convergence plus rapide mais non optimale de la solution. La deuxième étape du processus consiste à rajouter la contrainte d'unicité du robot. Cette étape revient à déplacer certaines opérations dans le planning afin de rendre toutes les opérations de transport réalisables avec un seul transporteur. Les auteurs ont montré que leur algorithme change beaucoup de dates de traitement, mais augmente très peu le makespan global. Ces deux étapes sont représentées dans la figure 27.

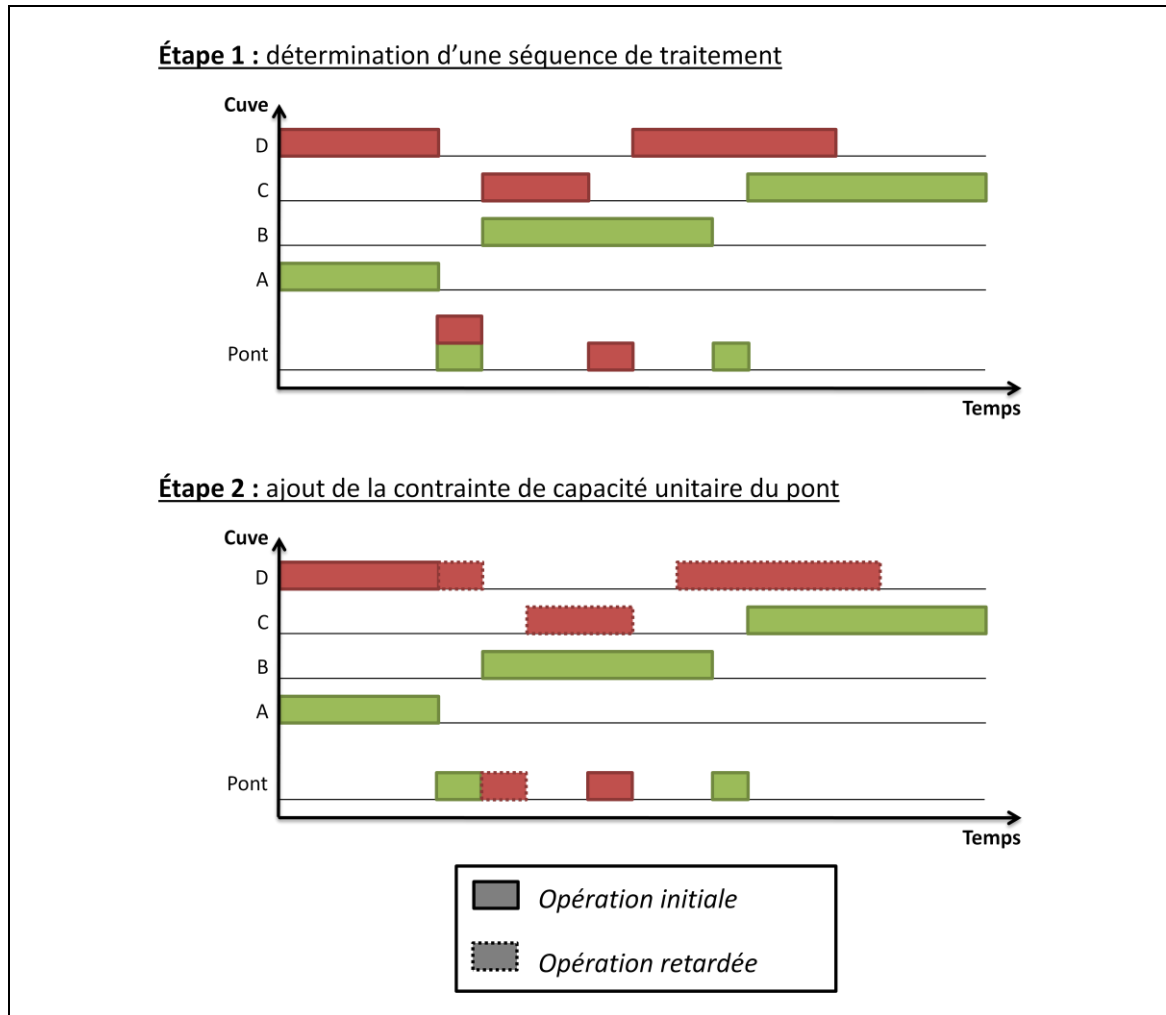


Figure 27 : Les deux étapes de [Bhushan S. et al., 2003]

Dans [Bhushan S. et al., 2004], les auteurs ont évalué différentes heuristiques pour améliorer les temps de calcul nécessaires à l'ordonnancement, à l'aide de programmation linéaire présentée dans l'article précédent. La *première partie* présente des algorithmes de placement des tâches une fois que celles-ci ont été séquencées.

Le premier algorithme exposé est nommé **JAT** (job-at-a-time). L'algorithme planifie les pièces les unes après les autres. Chaque pièce est planifiée complètement (de la première à la dernière opération) en intercalant les opérations de trempe et de transport dans les intervalles de temps disponibles du pont et des cuves, en cherchant la plus petite date de fin possible pour la tâche. L'état de la ligne doit alors être donné vide au début de la planification, et sera rendu vide à la fin de la planification. Cet algorithme, déjà utilisé dans les années 1980 par le logiciel *GenordOrdo*, est

relativement bon lorsqu'il y a peu de gammes de traitement, ou que celles-ci sont très semblables, mais il montre de réelles lacunes lorsque les gammes de traitement sont longues. En effet, la première pièce planifiée contraint le système, rendant souvent impossible l'insertion de la pièce suivante.

Le second algorithme proposé, prenant également en compte les opérations du pont, est basé sur le principe du **recuit simulé** (RS). De petites perturbations sont introduites dans un ordonnancement de base afin de déterminer une solution proche, éventuellement meilleure. La meilleure solution est alors soumise aux mêmes perturbations. L'amplitude des perturbations diminue avec le temps afin de converger vers un minimum local. L'ordonnancement est calculé à partir de l'algorithme JAT et les perturbations introduites sont des retards dans certaines opérations de transport afin d'espérer une meilleure imbrication des opérations, donc un meilleur *makespan*.

Le troisième algorithme d'optimisation par voisinage, appelé **iterative improvement** (II), est basé sur le principe du recuit simulé. La différence avec celui-ci réside dans le choix de la perturbation. Dans l'algorithme du recuit simulé, la perturbation est choisie de façon aléatoire, tandis que dans l'algorithme II, le choix est orienté afin de converger plus rapidement. La méthode de choix proposée favorise les grandes perturbations et impose une descente du *makespan* d'un pas sur l'autre.

Ces trois algorithmes ont été comparés à un algorithme proposé par Geiger, Kempf et Uzsoy dans [Geiger C.-D. *et al.*, 1997] et annoncé comme étant l'algorithme actuellement implanté dans les logiciels de pilotage des lignes de traitement de surface. Les résultats présentés montrent de meilleurs résultats (ou pour certaines instances, des résultats égaux) pour les algorithmes RS et II par rapport aux résultats de l'algorithme de Geiger.

La seconde partie présente deux stratégies pour le séquençement des tâches. Deux algorithmes, l'un basé sur la méthode **tabou** et l'autre sur le **recuit simulé**, ont été développés pour obtenir un séquençement des opérations. Ces deux méthodes de séquençement ont ensuite fait l'objet d'un couplage avec les méthodes de placement des tâches **iterative improvement** et de **Geiger**. Les résultats permettent évidemment de montrer l'utilité d'une méta-heuristique performante pour avoir des temps CPU acceptables lorsque la taille du problème devient importante, mais également, soulignent dans tous les cas l'importance de la solution initiale donnée à l'algorithme de descente. Enfin, le choix de la méthode de détermination de la séquence initiale dépendra également de la méthode de ré-ordonnancement choisie.

Dans [Zhu Y. *et al.*, 2004], les auteurs démontrent la NP complétude du problème d'ordonnancement des missions de déplacement des conteneurs dans un port (problème très similaire au problème de traitement de surface). Dans [Zhu Y. *et al.*, 2006], les auteurs ont comparé plusieurs méta-heuristiques utilisées pour l'ordonnancement des tâches. Leur étude montre que pour la recherche de la solution optimale, la résolution à l'aide de la méthode **branch and bound** est plus rapide que la résolution par programmation linéaire. De plus, la méta-heuristique du **recuit simulé**, plus rapide sur des problèmes de taille importante, fournit une solution très proche de l'optimum.

4.2. Méthodes en ligne

Dans le cas d'une demande non prévisible ou d'aléas dans les moyens de production, la gestion des aléas doit être prise en compte. Cette gestion ne peut se faire que par des méthodes de pilotage en ligne de la production.

J. Sgall, dans un chapitre dédié à l'ordonnancement en ligne [Sgall J., 1998], souligne la nécessité de l'ordonnancement en ligne, notamment pour son caractère réaliste. L'auteur met en avant l'impossibilité dans un problème industriel de connaître à l'avance tout le scénario de production. Pour ce faire, l'auteur soulève quatre paradigmes qui peuvent imposer l'utilisation d'un ordonnancement en ligne :

- **L'arrivée de produits au fil du temps.** Ce paradigme impose la planification des opérations des produits au fur et à mesure de leur arrivée dans le système de production.
- **L'absence de prévision des durées opératoires.** Dans ce contexte, la fin d'une opération n'est connue qu'au dernier moment. La prochaine opération ne peut alors être planifiée, qu'une fois la précédente réellement terminée.
- **Le retard sur l'arrivée des produits.** Dans ce cas, l'arrivée prévue des produits a été modifiée pour une raison externe, ce qui impose la prise en compte de cet aléa.
- **L'intervalle de planification.** Les trois paradigmes précédents permettent la prise en compte d'un aléa en cours de production. Ce 4^{ème} paradigme permet de modifier la date de livraison (donc la priorité) d'un produit.

Dans le cadre du pilotage en ligne des unités de traitement de surface, les travaux s'intéressent généralement aux trois premiers paradigmes afin de prendre en compte une demande non prévue et des durées variables sur certaines opérations.

Dans [Mangione F., 2003], l'auteur aborde le cas de l'ordonnancement cyclique mono-produit. Il souligne l'importance de la réactivité aux aléas, également dans ce type de problème. En effet, dit-il, l'ordonnancement cyclique optimise le rapport $\frac{\text{durée du cycle}}{\text{nombre de porteurs introduits durant un cycle}}$ sans prendre en compte les aléas de production ou les périodes transitoires. Un mode dégradé doit alors être développé pour l'absorption des aléas (souvent un décalage de tout le planning vers le futur). Dans le cas d'un changement de cycle, une des techniques serait de vider l'installation pour la remplir ensuite avec la nouvelle production. Une autre technique proposée dans [Varnier C., 1996] serait de découper la transition en quatre phases :

- Dégrader l'ancien cycle pour préparer l'insertion du nouveau cycle
- Insérer le nouveau cycle dégradé dans l'ancien cycle dégradé
- Vider l'installation des produits de l'ancien cycle
- Modifier le nouveau cycle dégradé pour le faire tendre vers son optimum

Le problème de la **gestion des conteneurs dans les ports** (appelé *crane scheduling*) et leur transport par les dockers se rapproche du problème du pilotage des unités de traitement de surface. En effet, les fenêtres temporelles présentes dans le traitement de surface se retrouvent dans les plages

courtes de disponibilité des navires ; les postes d'attente du traitement de surface se rapprochent des stations intermédiaires de chargement ; enfin la ressource critique des deux problèmes est le mécanisme de manutention. Dans ces deux problèmes, la demande peut être dynamique et les aléas fréquents, ce qui rend leurs méthodes de résolution relativement proches. Les problèmes de dépassement de fenêtres temporelles se traduisent d'un côté par une non qualité et de l'autre côté par un fort coût de monopolisation des navires. Une présentation détaillée de ce problème a été faite dans [Liebermana R.-W. *et al.*, 1981], avant de proposer un modèle dynamique de gestion du pilotage et des conflits des ponts dans le cas d'arrivées aléatoires des demandes de transfert.

4.2.1. Ordonnancement dynamique (pilotage avec prévision)

L'ordonnancement dynamique consiste à définir un séquençement prévisionnel des opérations. La séquence définie est exécutée jusqu'à l'arrivée d'une perturbation remettant en cause les prévisions. Les heuristiques développées pour résoudre le DHSP abordent séparément les perturbations engendrées par l'arrivée d'un produit et les perturbations engendrées par un aléa du système. L'arrivée d'un nouveau produit provoque la construction d'un nouveau planning de production tandis que l'occurrence d'un aléa aura pour effet un décalage (translation) de toutes les opérations à venir.

La méthode simple présentée dans [Yin N. *et al.*, 1992] consiste à insérer dans le planning existant les opérations de la nouvelle pièce. Dans le cas d'une insertion sans conflit, la nouvelle pièce est introduite dans le planning ; dans le cas contraire, l'entrée de la pièce est retardée. Cette méthode est développée dans le cas d'opérations de trempe sans tolérance. Elle montre de bons résultats dans le cas de petites gammes de traitement ou dans le cas où les durées opératoires sont bien supérieures aux durées de transport. L'application d'une telle méthode, en prenant en compte les tolérances sur les opérations de traitement, a été implantée dans les années 1980 dans un logiciel d'ordonnancement (*Génord*) réalisé par la société *Courbon*. L'exploitation de ce logiciel au niveau industriel a montré l'impossibilité de planifier de longues gammes (à partir de cinq opérations) même lorsque l'installation est peu chargée. L'utilisation des seules tolérances du nouveau produit ne suffit pas à permettre l'insertion de celui-ci dans le planning existant.

Les auteurs de [Yih Y., 1994] proposent un algorithme polynomial permettant l'insertion d'une nouvelle production dans un planning existant. L'amélioration par rapport à la méthode précédente réside dans la possibilité, pour l'algorithme exposé, de déplacer des tâches existantes (sans en changer l'enchaînement) pour insérer au plus tôt la nouvelle pièce. Cependant, l'algorithme montre une limite : le décalage d'une opération pour insérer une nouvelle pièce peut introduire des nouveaux conflits qui ne sont pas gérés. La gestion partielle des conflits peut alors générer un ordonnancement non réalisable.

Il existe d'autres algorithmes dynamiques qui remettent en cause tout l'ordonnancement existant (dates et enchaînements). Ces algorithmes sont souvent identiques à ceux développés dans l'ordonnancement prédictif en ajoutant la prise en compte de l'état courant non vide de l'installation. Dans [Chauvet F. *et al.*, 2000], les auteurs proposent une application de l'algorithme *branch and bound* pour l'ordonnancement en ligne assurant l'optimalité de la solution dans le cas où les durées de transport peuvent être négligées. Une extension de leur algorithme est proposée pour

traiter le cas où les durées de transport ne sont pas négligeables. Cette extension n'assure cependant pas l'optimalité du résultat. Quelques expérimentations montrent que l'algorithme est applicable en temps réel.

Le problème dynamique peut aussi être vu comme un problème d'ordonnancement où les durées de traitement sont des cibles souhaitées, mais non strictes. Dans [Fargier H. *et al.*, 2001], les auteurs appliquent une approche de résolution floue pour rechercher un ordonnancement des opérations de trempe. L'ordonnancement des opérations est remis en cause à chaque nouvelle arrivée de produit. Dans la méthode proposée, les intervalles de traitement ne sont pas imposés par l'ordonnanceur, mais la qualité des produits à la sortie de l'ordonnancement est évaluée. Cette qualité est estimée pour chaque produit. Elle correspond au pire traitement qu'il a subi (dépassement le plus important). Cette méthode a l'avantage, dans un *pilotage en ligne*, de trouver un ordonnancement dégradé dans le cas où l'état courant de l'installation n'est plus compatible avec les contraintes du système. Cependant, elle n'assure pas la stricte appartenance des temps de traitement aux fenêtres temporelles imposées.

Dans [Fleury G. *et al.*, 1999] puis dans [Fleury G. *et al.*, 2001], les auteurs ont traité la prise en compte des aléas lors de la réalisation d'un planning pour le pilotage des unités de traitement de surface. Le planning créé a pour caractéristique d'être peu sensible à de faibles perturbations. Cependant, même si de faibles perturbations peuvent être prises en compte dans le planning, ce mode de pilotage ne couvre que le deuxième et le troisième paradigme de [Sgall J., 1998]. L'arrivée de perturbations plus importantes imposera le ré-ordonnancement dynamique de la production. La méthode proposée pour la réalisation d'un planning permettant de prendre en compte les aléas est validée par simulation stochastique. La méta-heuristique proposée permet de limiter le nombre de ré-ordonnancements.

4.2.2. Ordonnancement réactif (pilotage sans prévision)

Par définition, le pilotage réactif ne prévoit pas d'opération à l'avance. Cette caractéristique l'empêchant de garantir la qualité des traitements le rend très délicat à appliquer. Les travaux qui abordent cette voie sont très rares, et leurs résultats, bien que scientifiquement intéressants, ne sont appliqués que dans des installations où les tolérances sont très larges ou dans le cas d'acceptation des dépassements des durées de trempe. C'est pour cette raison que très peu d'études ont été menées sur le RHSP.

On peut néanmoins classer les méthodes de pilotage réactif des installations de traitement de surface en deux catégories.

La première catégorie regroupe les **méthodes par apprentissage**. Dans ces méthodes, un système de pilotage actuel est utilisé et les décisions prises sont évaluées et capitalisées. Lorsque le système de production se retrouve devant une décision à prendre, le système de pilotage compare l'état courant aux états de sa banque de données et applique une décision déjà connue. Dans [Sammur C., 1996], l'auteur propose une méthode générique de *construction automatique de règles* pour le pilotage réactif basé sur l'analyse des décisions prises par le système de pilotage actuel. Cette méthode est utilisée pour transférer la connaissance d'un humain expert pilotant un système de production vers

un système de pilotage réactif automatisé. Dans [Yih Y., 1990], l’auteur propose de capitaliser les données par analyse de la trace du système (*TDKA, trace-driven knowledge acquisition*).

La seconde catégorie regroupe les méthodes basées sur la **définition de règles**. Les auteurs de [Rebollo M. *et al.*, 2000] proposent une méthode basée sur les systèmes multi-agents pour l’ordonnancement des missions de transport dans les terminaux portuaires. Le *chapitre 3* de cette thèse expose un pilotage réactif et ses limites intrinsèques.

4.3. Bilan sur les méthodes existantes

Le problème d’ordonnancement cyclique a des caractéristiques propres et des méthodes de résolution spécifiques évoquées précédemment. De leur côté, les problèmes prédictifs et dynamiques ont été abordés à l’aide des méta-heuristiques et d’algorithmes performants. La comparaison des différentes méthodes par les scientifiques montre que, sur ce type de problème à fenêtre temporelle stricte, l’algorithme **branch and bound** donne de très bons résultats.

Habituellement, lorsque l’exploration des solutions montre un nœud dont les performances sont inférieures à la borne (*bound*), l’algorithme remonte au nœud parent pour continuer l’exploration. Lamothe *et al.*, dans [Lamothe J. *et al.*, 1995], proposent une fonction d’exploration plus performante, permettant d’éliminer les nœuds enfants ainsi que certains nœuds du même niveau dans le cas où les performances du nœud évalué sont bien inférieures à la borne. Le gain en temps de calcul est supérieur à 90% (sur les cas présentés) et atteignent le même résultat.

Les différentes études présentées ici mettent en avant la facilité de trouver une borne maximale qui permet l’élimination de grandes sous-parties du champ des solutions. Enfin, le pilotage réactif n’est que rarement utilisé. L’impossibilité de garantir les durées de traitement le rend inapplicable dans le cas où les dépassements (même courts) ne sont pas tolérés.

5. Conclusion

Le coût d’une installation de traitement de surface est relativement important pour des raisons de dimensions de l’atelier (une cuve peut mesurer 20 mètres de long par 3 mètres de large et 5 mètres de haut), pour des raisons de moyens de production (les bacs contiennent plusieurs dizaines de mètres cubes de produits onéreux), pour des raisons écologiques (ces installations font partie des installations *‘polluantes’*). La rationalisation des moyens de production passe par une production en continu. L’optimisation de l’implantation fait partie du processus d’optimisation global du moyen de production, au même titre que l’optimisation du module de pilotage.

On peut séparer le problème global de l'optimisation d'un moyen de production, en trois sous problèmes :

1. **Le moyen physique.** Cette étape passe par la modification de la constitution de l'installation (ajout, suppression de ressources) ou la modification de l'agencement des moyens mis en œuvre.
2. **Le système de pilotage.** La modification du système de pilotage consiste à modifier l'architecture, le mode de pilotage ou les algorithmes mis en place. Ces modifications, en entraînant des changements dans le système informationnel, influent sur la production.
3. **La production.** L'ordre de prise en compte des commandes, leur affectation sur les sites de production ou la sous-traitance de certaines commandes influent sur les performances du moyen de production. Cette réflexion située en amont de la production, sans modifier les commandes, permet une meilleure optimisation globale des moyens disponibles.

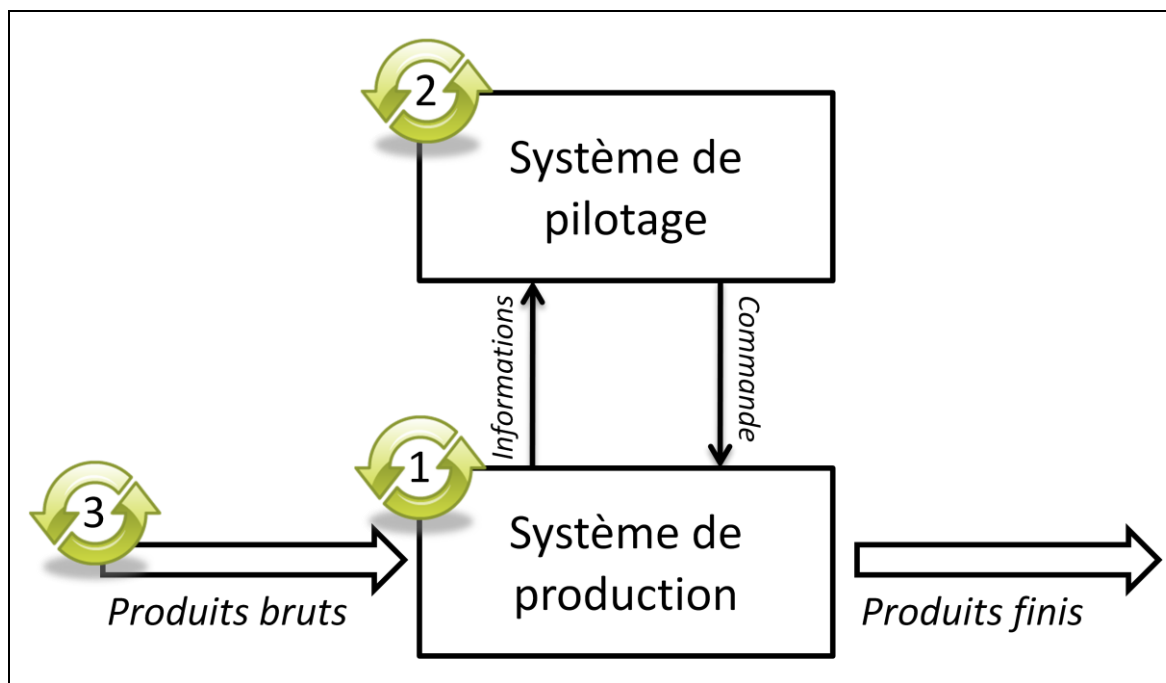


Figure 28 : Les trois optimisations d'un système de production

Nous proposons d'étudier ces trois optimisations dans le cas du pilotage réactif des unités de traitement de surface. Pour chacune des optimisations, nous proposerons une méthode de résolution. L'optimisation du système de pilotage sera étudiée dans le chapitre 3 ; l'optimisation du système de production sera abordée dans le chapitre 4 ; enfin, l'optimisation du lancement des produits sera traitée dans le chapitre 5.

Chapitre 3 : Couplage entre pilotage réactif et pilotage prédictif

| | |
|--|-----------|
| 1. Présentation du problème | 56 |
| 1.1. Topologies étudiées | 56 |
| 1.2. Méthodologie pour la définition du pilotage réactif | 57 |
| 1.3. Présentation des cas industriels | 58 |
| 2. Proposition d'un pilotage réactif | 60 |
| 2.1. Principe du pilotage | 60 |
| 2.2. Définition des règles | 61 |
| 2.3. Modèle de simulation | 65 |
| 2.4. Exploitation des simulations | 71 |
| 2.5. Analyse des résultats | 74 |
| 3. Couplage entre pilotage réactif et prédictif | 76 |
| 3.1. Bref état de l'art | 76 |
| 3.2. Principe du couplage | 76 |
| 3.3. Modèle holonique | 78 |
| 3.4. Application à un cas industriel | 79 |
| 4. Implémentation | 80 |
| 4.1. Choix d'un outil d'ordonnancement | 80 |
| 4.2. Pilotage réactif ou pilotage dynamique ? | 81 |
| 4.3. Solution adoptée | 83 |
| 5. Conclusion | 83 |

Nous avons évoqué dans le premier chapitre quelques évolutions des systèmes organisés, et notamment l'influence de la personnalisation de masse sur les modes de pilotage d'atelier. Nous avons exposé quelques modes de pilotage qui permettent une bonne réactivité et une meilleure flexibilité. Dans le deuxième chapitre, nous avons présenté une problématique de pilotage industrielle : le pilotage des lignes de traitement de surface. Nous y avons développé une classification des problèmes ainsi qu'un aperçu des modes de pilotage existant dans la littérature.

Ce troisième chapitre rapporte une grande partie des travaux de cette thèse. Après avoir soumis le problème de traitement de surface étudié et sa situation au sein des HSP, nous proposerons un mode de pilotage purement réactif. Nous analyserons les résultats d'un tel pilotage et en extrairons quelques lacunes insolubles en pilotage réactif. Nous proposerons alors un couplage entre ce mode purement réactif et une couche prévisionnelle permettant de sécuriser les opérations critiques.

1. Présentation du problème

1.1. Topologies étudiées

En vue de permettre la réutilisation des travaux présentés ici pour un grand nombre de problèmes, de topologie, de taille d'installation, de type de production... il est nécessaire d'effectuer le moins d'hypothèses possibles. Le tableau 7 résume les hypothèses faites dans la suite de ces travaux. Ces hypothèses sont liées à la classification présentée dans le chapitre précédent.

| Critère | Typologie |
|--|--|
| Le nombre de ligne n'a pas d'importance. | $\beta_1 \in \mathbb{N}^*$ |
| Le nombre de <i>navette</i> (mécanisme de transfert) et leur position n'a pas d'importance. | $\beta_2 \in \mathbb{N}$ |
| Les transporteurs et les navettes peuvent fonctionner de manière synchronisée ou non. | $\beta_3 \in \{\emptyset, \text{synchro}\}$ |
| Le nombre de transporteurs, le nombre de bays par ligne et le nombre de places par bay n'est pas important. | $\beta_{4,1} ; \beta_{4,2} ; \beta_{4,3}$ quelconques |
| Les porteurs peuvent être gérés ou non. | $\beta_5 \in \{\emptyset\} \cup \mathbb{N}^*$ |
| La gestion des porteurs vides encombrants ne sont pas gérés. | $\beta_8 = \emptyset$ |
| Le mode de gestion des postes de chargement et déchargement n'a pas d'impact. | $\beta_9 = \emptyset$ |
| Le pilotage est capable d'ordonnancer un très grand nombre de produit (complexité d'ordre 1 en fonction du nombre de produits) | $\delta_1 = \infty$ |
| Le nombre de gammes n'a pas d'impact sur les calculs. | $\delta_2 = \infty$ |
| Le nombre d'opérations de la plus longue gamme n'a pas d'influence sur les calculs. | $\delta_3 = \infty$ |
| Les porteurs peuvent être nettoyés après déchargement. Il suffit d'ajouter dans la gamme une telle opération. | $\delta_4 \in \{\emptyset, \text{clean}\}$ |
| Les bays peuvent être mono ou multi fonctions. | $\delta_5 \in \{\emptyset, \text{recr}\}$ |
| L'objectif du pilotage est la maximisation de la productivité. | $\gamma = \text{productivité}$ |

Tableau 7 : Typologies des implantations abordées

De plus, il ne sera pas fait d'hypothèse sur le schéma d'implantation de l'installation de traitement de surface. La méthode de pilotage présentée permettra, sans modification, de piloter des lignes en U, en O, en H, ou des lignes à topologie combinée.

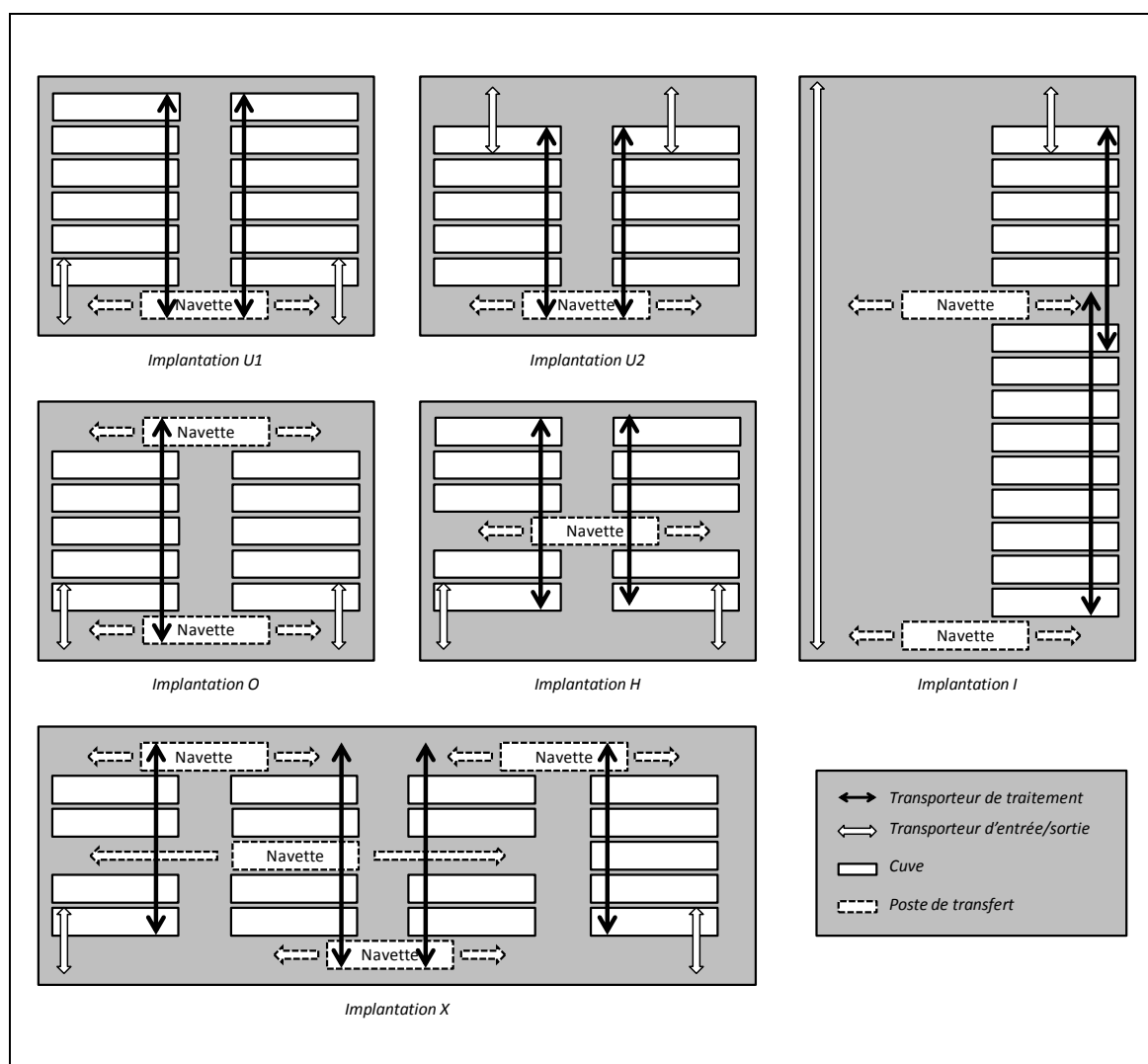


Figure 29 : Exemple d'implantations abordées

La figure 29 présente quelques schémas d'implantations d'installations pilotables à l'aide de ces travaux. Dans le cas d'ordonnancement multi-pont, chaque pont sera géré sur une zone bien définie, avec la possibilité de partager une petite partie de cette zone avec un autre pont. C'est par exemple le cas de l'implantation I.

1.2. Méthodologie pour la définition du pilotage réactif

Nous avons défini, dans le chapitre 1, le concept de pilotage réactif. Dans le chapitre 2, nous avons décrit la notion d'ordonnancement réactif. Dans ce mode de pilotage nous ne faisons pas de prédiction. Le comportement du système est la conséquence de l'application de règles locales. Si cette approche du pilotage est intéressante par son aptitude à traiter des systèmes complexes et de grande envergure, elle soulève le délicat problème de la définition des règles de comportement.

Notons que cette vision du pilotage est proche du paradigme informatique des systèmes multi-agents. Le comportement global du système émerge à partir du comportement local d'agent réactif, lui-même basé sur l'application de règles simples. Il nous paraît donc logique, pour définir une méthodologie de conception d'un pilotage réactif, de nous tourner vers les méthodologies de conception des systèmes multi-agents. Comme le soulignent [Sabas A. *et al.*, 2001] et [Sabas A. *et al.*, 2002] « *Plusieurs méthodologies ont été proposées pour le développement des SMA. Ces méthodologies constituant soit une extension des méthodologies orientées-objet, soit une extension des méthodologies à base de connaissances, demeurent incomplètes.* ». Un exemple d'approche méthodologique qui nous paraît intéressante est la méthode Cassiopée proposée par [Collinot A. *et al.*, 1996]. Cette méthodologie repose sur le principe de parcimonie : le modèle de l'agent est construit de façon incrémentale, en couches successives, jusqu'à ce que le système réalise la fonction demandée. L'intérêt principal de l'approche Cassiopée est qu'elle est ascendante, bien adaptée à la définition d'agents réactifs. Elle propose, dans un premier temps de définir les agents et leur rôle. Ensuite, dans un deuxième temps, elle propose de chercher à définir le comportement des agents. Nous avons, dans ce cadre, défini progressivement des règles de fonctionnement, en testant leur comportement par une démarche de simulation. Cette approche a été utilisée par [Breton L. *et al.*, 2004] pour concevoir un système de pilotage réactif d'AGVs (chariots autoguidés).

1.3. Présentation des cas industriels

Les travaux constitutifs de cette thèse ont été appliqués à deux problèmes industriels issus du milieu de l'aéronautique. Cette application permet, d'une part, la validation des concepts scientifiques développés dans ces travaux et, d'autre part, leur transfert technologique vers l'industrie.

La première installation étudiée (appelée **installation industrielle A**, voir figure 30) est constituée de deux lignes de traitement implantées en parallèle (implantation de type *U2*, voir figure 29). Chacune des lignes dispose de 5 postes d'entrée/sortie et d'une vingtaine de cuves (cuve de traitement, position d'attente, cuve pour opération manuelle). Chacune de ces lignes est desservie par un pont unique. Les lignes de droite (LD) et de gauche (LG) sont reliées entre elles par un poste de transfert (utilisé actuellement dans un seul sens). Cette unité est capable de traiter une vingtaine de gammes. Les gammes peuvent :

1. rentrer par LD et sortir par LD.
2. rentrer par LD, passer par le transfert et sortir par LG.
3. rentrer par LG et sortir par LG.

L'installation A tourne en permanence et dispose d'une zone de stockage externe des porteurs préparés ou vides. La rentrée et la sortie des porteurs dans la zone 'entrée/sortie' sont gérées manuellement. L'étude se limite à l'ordonnancement des opérations entre les postes de chargement et de déchargement. Selon la nomenclature de référence, ce problème est :

$$RSHP \mid 2,1, synchro, (1,20, \emptyset), (1,20, \emptyset) // ass \mid 500/100,25, \emptyset, \emptyset \mid C_{max}$$

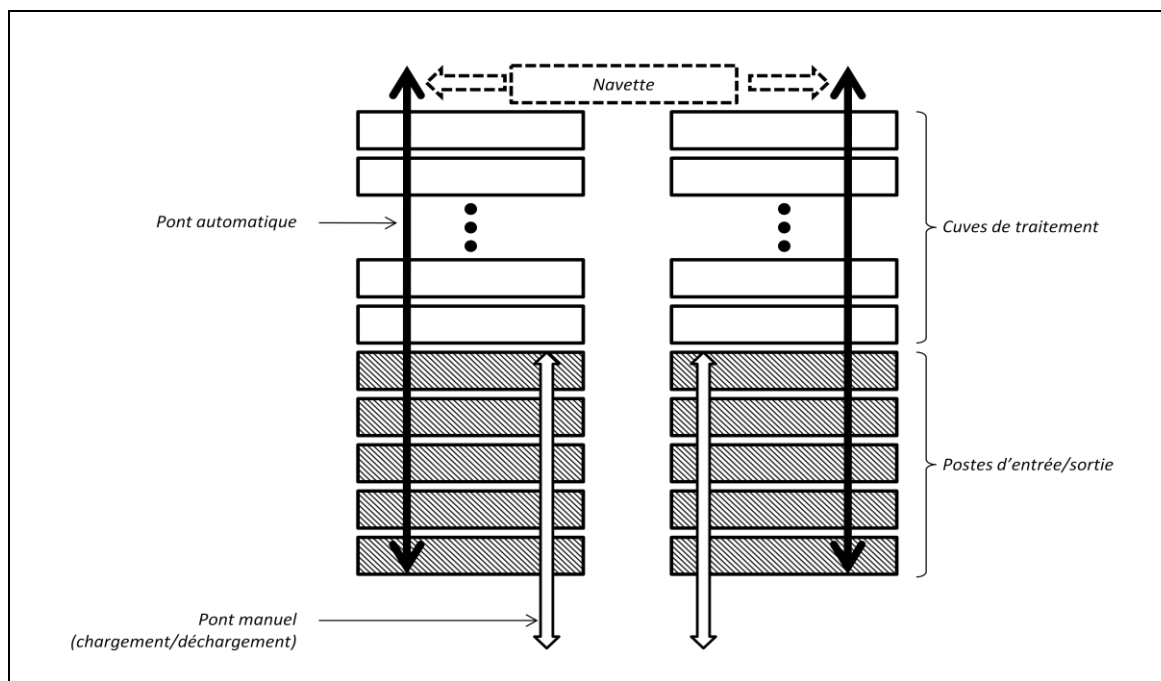


Figure 30 : Installation industrielle A

La seconde installation étudiée (appelée **installation industrielle B**, voir figure 31) est une installation en cours de conception. La topologie des nouveaux locaux et le matériel disponible imposent une implantation en I avec deux lignes l'une derrière l'autre. Cette installation aura pour objectif d'obtenir une bonne productivité en réduisant le nombre de gammes prises en compte (3 ici). La réduction du nombre de gammes pourrait permettre de se rapprocher des méthodes de résolution des CHSP, mais l'arrivée non prévue de pièces prioritaires limite l'intérêt d'un pilotage a priori.

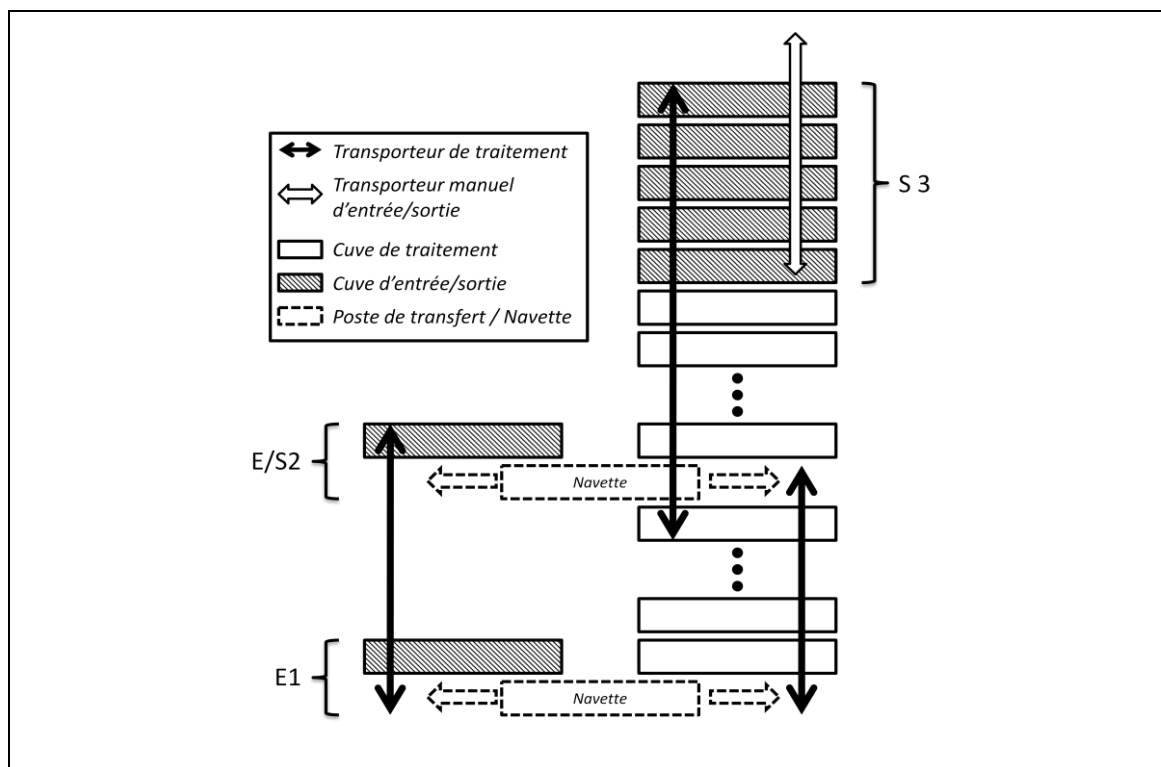


Figure 31 : Installation industrielle B

Cette installation est constituée de deux transporteurs assurant le déplacement des pièces. Ces transporteurs se déplacent sur le même chemin de roulement. Les transporteurs ont la capacité mécanique de se déplacer sur toutes les cuves de la ligne de droite. Les pièces peuvent arriver par E1 et ressortir par E/S 2 ou S3, ou rentrer par E/S2 et sortir par S3. L'arrivée des pièces sur les postes d'entrée sortie et le recyclage des palonniers vides ne sont pas pris en compte dans le pilotage automatique. Selon la nomenclature de référence, ce problème est le suivant :

$$RSHP \mid 2,2, \text{synchro}, (1,4,1), (2,20, \emptyset) // \text{ass} \mid 500/3,25, \emptyset, \emptyset \mid C_{max}$$

2. Proposition d'un pilotage réactif

La première partie de cette proposition consiste à développer une heuristique pour le pilotage purement réactif. Le principe général ainsi que les règles détaillées seront présentés ci-après.

2.1. Principe du pilotage

Le pilotage réactif proposé est basé sur le principe client/serveur ou demandeur/fournisseur, repris dans la figure 32.

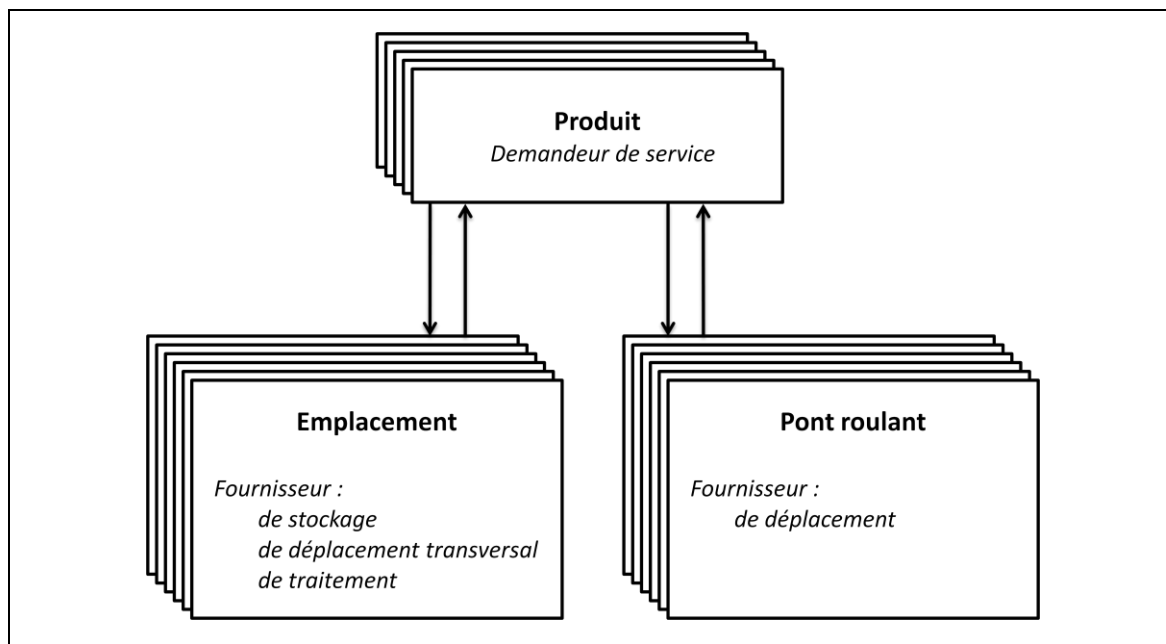


Figure 32 : Architecture client/serveur appliquée au RHSP

Les clients de cette architecture représentent les produits à traiter, ceux-ci devant subir différentes transformations, elles-mêmes réalisées dans les bains de traitement. Pour ce faire, ils effectueront deux types de requêtes : des demandes de traitement, effectuées auprès des cuves de traitement, et des requêtes de transport, effectuées auprès de ponts.

Les serveurs sont de deux types, classés en fonction des services qu'ils peuvent fournir. D'un côté les emplacements sont capables de fournir des *services* de stockage, traitement, déplacement

transversal (navette). D'un autre côté les ponts roulants sont capables de fournir un transport. Dans le mode de pilotage que nous proposons, tous les serveurs seront à capacité unitaire, mais certains pourront être dupliqués afin de réduire l'impact des postes goulots. Ainsi plusieurs postes de stockage d'attente ou plusieurs cuves de traitement identique pourront exister simultanément.

La dynamique du système s'inspire ensuite du protocole *contract-net* présenté dans [Smith R.-G., 1980]. Lors d'un besoin de la part du client, celui-ci interroge les différents fournisseurs de services pouvant répondre à sa demande, il liste les réponses positives et choisit le meilleur fournisseur. Le client effectue alors la réservation du fournisseur sélectionné. De son côté, le fournisseur informe les clients de sa capacité à répondre à leur besoin, reçoit un ensemble de demandes et y répond dans un ordre qu'il établit selon des règles internes présentées dans le paragraphe suivant.

2.2. Définition des règles

Les règles correspondent aux mécanismes de décision utilisés par les clients, pour choisir leur fournisseur, et par les fournisseurs, pour ordonner les demandes des clients. On trouvera alors des règles liées aux produits, et des règles liées aux fournisseurs de service.

2.2.1. Règles produit/emplacement

a. Produit sur un poste d'entrée

Règle R1.1 : Un produit est sur un poste d'entrée/sortie ne peut effectuer une demande de déplacement vers un poste, que dans le cas où la charge de la ligne (du poste destinataire) est inférieure à une limite.

Règle R1.2 : Si la charge est inférieure à la capacité, le produit recherche une cuve capable d'effectuer le premier traitement. Si plusieurs cuves sont capables d'effectuer le traitement, le produit sélectionne la cuve la plus proche. Le produit réserve la cuve et effectue une demande de déplacement au pont.

La limite de capacité de chaque ligne fait partie des paramètres à régler après analyse du comportement émergent. Une limite haute aura pour conséquence un encombrement de l'installation, donc des dépassements importants sur les durées de traitement. Une limite basse aura pour conséquence une baisse de la productivité.

b. Produit sur un poste de transfert

Règle R2 : Un produit sur un poste de transfert n'est plus comptabilisé dans la charge de la ligne en amont, et devient comptabilisé dans la charge de la ligne en aval. Dans ce cas seulement, la charge d'une ligne peut être supérieure à sa limite (voir R1). Un produit sur un poste de transfert situé à côté de la ligne aval effectue les mêmes opérations qu'un produit en poste d'attente.

c. Produit sur un poste d'attente

Règle R3 : Lorsqu'un produit est sur un poste d'attente, il établit la liste des cuves capables d'effectuer le prochain traitement. Dès qu'une cuve est/devient libre, il réserve la cuve, puis il effectue une demande de déplacement au pont.

d. Produit sur un poste de traitement

Règle R4 : Lorsqu'un produit est en cours de traitement, on définit deux dates caractéristiques du traitement en cours : $tAnticipation_{b,o}$ et $tDegagement_{b,o}$:

$$\begin{aligned} tAnticipation_{b,o} &= tDeb_{b,o} + dMin_{g,o} - dAnticipation_p \\ tDegagement_{b,o} &= tDeb_{b,o} + dMax_{g,o} - dLimite_p \end{aligned}$$

On identifie alors trois périodes pendant un traitement :

1. de $tDeb_{b,o}$ à $tAnticipation_{b,o}$, le traitement s'effectue sans aucune tentative de prise de décision.
2. de $tAnticipation_{b,o}$ à $tDegagement_{b,o}$, le produit établit la liste de cuves capables de fournir le traitement suivant et essaye d'en réserver une. Si le produit arrive à réserver une cuve, il effectue, au pont, une demande de transfert vers cette cuve.
3. à partir de $tDegagement_{b,o}$, le produit établit la liste des positions d'attentes capables de le recevoir (= position sur la même ligne et disponible). Si un poste d'attente est libre, le produit le réserve et effectue, au pont, une demande de déplacement.

Cette règle évite les inter-blocages (produit en A voulant aller en B tandis qu'un produit en B désire aller en A, ou exemple de la figure 19). L'un des produits sera alors déplacé vers un poste d'attente.

Les paramètres $dAnticipation_p$ et $dLimite_p$ sont réglés grâce à des essais successifs effectués par simulation.

e. Cas de réservation multiple

Règle R5 : Dans le cas où l'on va vers un bain critique ($Crit_{g,o} = Vrai$), le poste est considéré comme libre s'il peut réserver son rinçage associé. La réservation du bain critique induit la réservation du bain de rinçage associé. Dans le cas de plusieurs bains critiques successifs, la réservation multiple s'effectue en chaîne.

Cette règle assure la disponibilité du rinçage en fin de traitement. Par contre, en ne permettant, à un instant donné, de n'avoir qu'un seul produit dans l'ensemble bain+rinçage, on limite la productivité du système.

2.2.2. Règles produit/pont

a. Choix de la mission

Règle R6 : Les missions reçues par le pont sont mises en concurrence les unes avec les autres selon les règles suivantes :

1. Chaque poste est caractérisé par une priorité. Une mission provenant d'un poste plus prioritaire sera traitée avant une mission reçue d'un poste moins prioritaire. La priorité d'un poste dépend principalement du caractère critique de l'opération qu'il fournit : un poste d'attente ou d'entrée ne sera pas prioritaire ; un traitement de rinçage sera de priorité moyenne ; un traitement critique sera de priorité élevée.
2. Les missions provenant de cuves de traitement (critique ou non) sont caractérisées par une date de sortie au plus tard. Le pont effectue les déplacements demandés en fonction de la date de fin au plus tard de l'opération en cours, en commençant par les plus petites dates.
3. En cas d'égalité des critères, le pont effectue la première mission reçue (Règle FIFO : First In First Out).

La durée d'anticipation des demandes ($dAnticipation_p$) peut avoir pour conséquence de faire arriver le pont, avant la fin d'un traitement, au-dessus de la cuve contenant la pièce à déplacer. Dans ce cas, le pont attend le temps nécessaire à la fin du traitement de la pièce mais ne remet pas en cause la mission choisie.

b. Traitement court

Règle R7 : Chaque bain est caractérisé par un paramètre $dCourt_p$. Si la durée de traitement du palonnier déposé est inférieure à ce paramètre ($dMini_{b,o} < dCourt_p$), le pont attend en position basse et n'accepte aucune autre mission que celle de sortie du palonnier.

Cette règle limite le nombre de dépassements lors d'un traitement court.

- Un bain de courte durée de traitement, mais acceptant une large tolérance pourra être réglé avec : $dCourt_p = 0$.
- Un bain imposant de conserver le pont sur place pour tous les traitements, pourra être réglé avec : $dCourt_p = \infty$.

c. Conflit de ponts

Règle R8 : Dans le cas où deux ponts se partagent une zone commune (par exemple dans l'implantation industrielle B présentée précédemment), les ponts effectuent normalement leur choix de mission et commencent leur déplacement.

- Au moment d'arriver à l'entrée de la zone commune, le 1^{er} pont réserve cette zone qui devient indisponible pour le 2nd pont.
- Dans le cas où cette zone est indisponible, le 2nd pont attend la libération (par le 1^{er} pont) de la zone commune.
- Dans le cas où un pont se trouve dans la zone commune sans mission, il se déplace vers une position de repli pour en sortir au plus vite.

Cet algorithme est un des plus simples dans la gestion des conflits de deux ponts. La zone commune est définie à l'avance en fonction des critères physiques des ponts (écartement minimum entre deux ponts). Dans les essais que nous avons réalisés, cet algorithme a été suffisant.

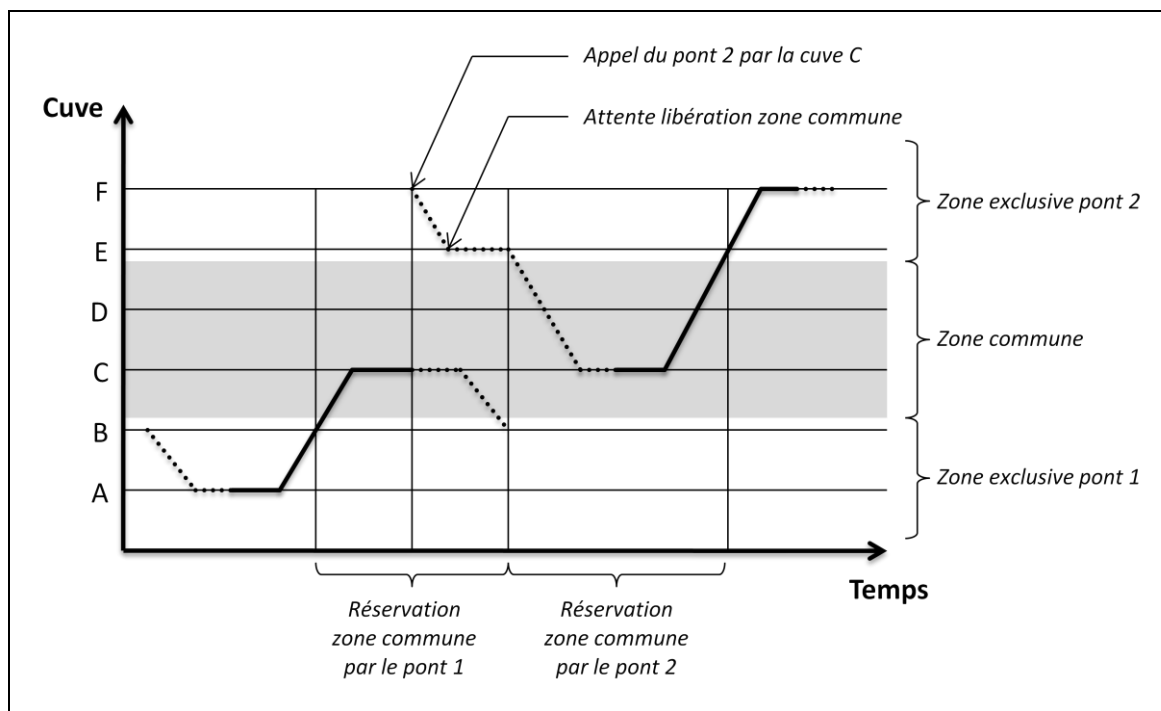


Figure 33 : Transfert d'une pièce à travers une zone commune

La figure 33 représente les mouvements de deux ponts (pont 1 et pont 2) pour l'échange d'une pièce à travers une zone commune. Le détail des actions de chaque entité est repris dans le tableau 8. Cet enchaînement des opérations est similaire à celui d'un échange à travers une navette de transfert.

| | Cuve A | Pont 1 | Cuve C | Pont 2 | Cuve F |
|----|---|---|--------|--------|--------|
| 1. | La cuve A dispose d'une pièce à déplacer vers la cuve F. Elle réserve une cuve de la zone commune et effectue une demande de déplacement au pont. | | | | |
| 2. | | Le pont reçoit la demande, va chercher la pièce, rentre dans la zone commune et dépose la pièce en C. | | | |

| | Cuve A | Pont 1 | Cuve C | Pont 2 | Cuve F |
|----|--------|--|---|---|--|
| 3. | | Le pont 1 se retire vers sa position de repli puis libère la zone commune. | La cuve C reçoit la pièce et effectue une demande de déplacement au pont 2. | | |
| 4. | | | | Le pont 2 commence le déplacement vers la cuve C et attend à l'entrée de la zone commune la libération de celle-ci. | |
| 5. | | | | La zone commune est libre, le pont y pénètre et prend la pièce qu'il va transporter vers la cuve F. | |
| 6. | | | | | La cuve F reçoit la pièce et effectue le traitement. |

Tableau 8 : Transfert d'une pièce à travers une zone commune

2.3. Modèle de simulation

La validation d'un pilotage réactif passe par la simulation de celui-ci avec des données réelles de production. Nous avons décidé, afin de permettre de valider ce mode de pilotage sur plusieurs topologies différentes (notamment les deux topologies industrielles présentées), de réaliser une implémentation des différentes règles sous forme d'objets. Chaque objet correspond à un élément physique ou virtuel, et est muni des règles qui le concernent. L'implémentation a été réalisée avec le logiciel de modélisation et de simulation des systèmes à événements discrets **Siman/Arena**⁷.

2.3.1. Objets physiques

- **Les ponts roulants** : les ponts roulants reçoivent les missions des positions (entrée/sortie, bails, attente, transfert) dont ils sont responsables, et classent les demandes suivant les règles définies précédemment. Ils s'occupent de tous les déplacements et gèrent les conflits

⁷ Logiciel de la société Rockwell Software

avec les autres ponts. Il est possible de les bloquer régulièrement pour simuler une panne et analyser l'impact d'un aléa sur le planning final. Ils sont caractérisés par :

- Une matrice de temps de déplacement : $DeplV_{p1,p2}$.
- Un générateur de panne (loi de fréquence, loi de durée).
- La limite du nombre de palonniers de la ligne.
- Une liste de postes d'attente.
- Un poste de repli en cas d'encombrement en zone partagée.

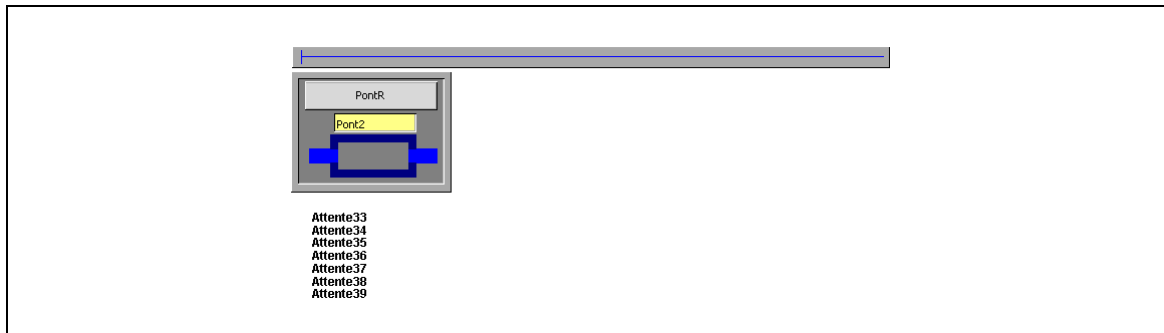


Figure 34 : Bibliothèque Siman/Arena : Pont

- **Les bains** : ces objets représentent les cuves de traitement ou de rinçage. Ils effectuent le chronométrage de traitement, anticipent les demandes de déplacement, décident le dégagement en attente. Ils sont caractérisés par :
 - Un traitement réalisable.
 - Deux voisins, pour déduire la topologie.
 - Un pont de prise. C'est à lui que la cuve effectuera ses demandes. En cas de besoin d'un poste d'attente, la cuve utilisera la liste des postes d'attentes accessibles à ce pont.
 - Une durée de traitement court : $dCourt_p$.
 - Une durée d'anticipation des demandes au pont : $dAnticipation_p$.
 - Une durée limite avant décision de dégagement en attente : $dLimite_p$.
 - Une criticité : utilisée par le pont pour ordonner les demandes reçues.
 - Un booléen indiquant sa situation en zone partagée par plusieurs ponts. Ce paramètre sert à effectuer des demandes de repli à un pont inactif sur la cuve.
 - Un booléen indiquant la nécessité de réserver le bain suivant au moment de la réservation de ce bain. Ce booléen est utilisé par le bain précédent dans le mécanisme de réservation en cascade (avant d'effectuer une demande de déplacement).

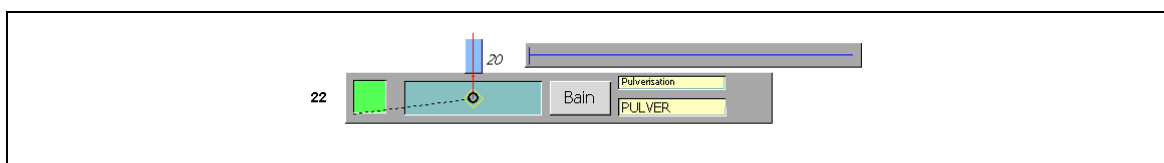


Figure 35 : Bibliothèque Siman/Arena : Bain

- **Les postes d'attente :** ils ont les mêmes caractéristiques et les mêmes mécanismes de réservation que les bains, à l'exception des paramètres liés à un traitement :
 - Deux voisins.
 - Un pont de prise.
 - Un booléen indiquant sa situation en zone partagée par plusieurs ponts

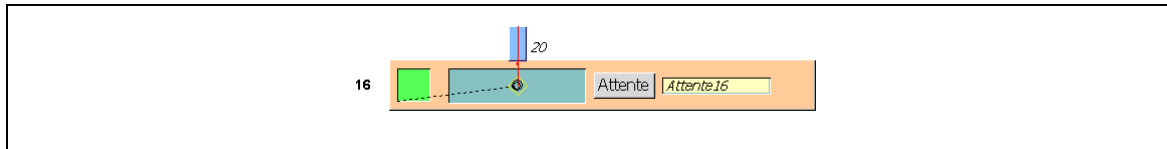


Figure 36 : Bibliothèque Siman/Arena : Attente

- **Modules de lancement :** les modules de lancement sont constitués de 4 postes d'entrée et 1 poste de sortie. Les postes d'entrée reçoivent des pièces brutes du module de répartition (voir plus loin) et réagissent comme un poste d'attente. Ils gèrent la limite de capacité du pont roulant auquel ils sont associés, afin d'éviter une demande de déplacement qui ferait dépasser la capacité de la ligne. Le poste de sortie ne sert que de poste de dépose pour les produits ayant fini leur traitement. Une modélisation plus fine du comportement de ce poste permettrait de dimensionner le nombre de poste de décrochage dans l'installation.

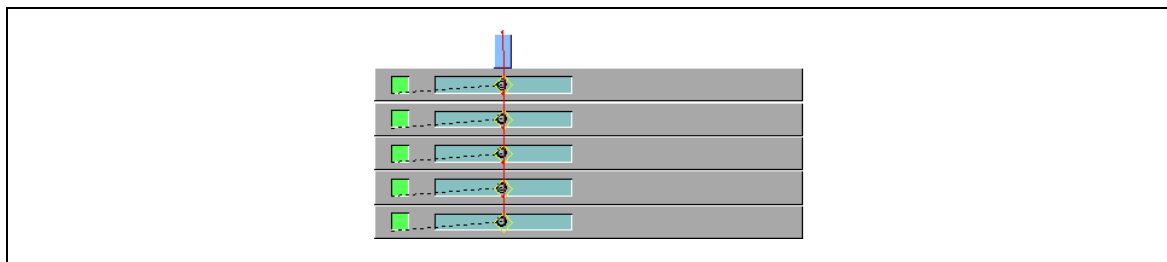


Figure 37 : Bibliothèque Siman/Arena : Lancement

- **Poste de transfert :** ces postes, utilisés pour synchroniser deux lignes parallèles, fonctionnent globalement comme un poste d'attente. Une fois libre, un bain peut demander un déplacement de sa pièce vers le poste de transfert qui reçoit la pièce, la déplace et effectue une demande au pont du deuxième côté. Il faut ajouter aux mécanismes d'une attente, les déplacements transversaux. Ce type de poste est donc caractérisé par :
 - Un pont de prise de chaque côté.
 - Des durées de déplacement.
 - Un booléen d'appartenance à une zone commune pour chaque côté.
 - Un côté de repli en cas d'inactivité.

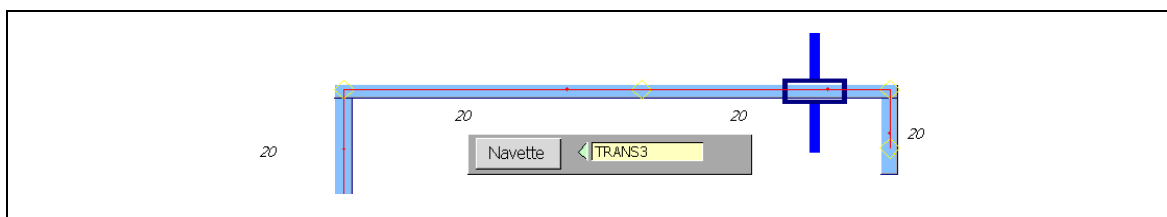


Figure 38 : Bibliothèque Siman/Arena : Navette

2.3.2. Objets virtuels

Il faut ajouter aux objets physiques des objets virtuels permettant de décrire le scénario simulé. La bibliothèque **Siman/Arena** est alors munie de 3 objets supplémentaires :

- **Les gammes** : permettant de décrire l'enchaînement des opérations. Modélisées à l'aide de séquences sous Siman/Arena, elles sont constituées d'une succession d'étapes caractérisées par :
 - Le type de traitement.
 - La durée minimum.
 - La durée maximum.
 - La durée d'égouttage.
 - Les durées de manipulation.
 - L'interdiction de passer par une attente à certains stades de traitement.
- **Le module de lancement** : cet objet permet de gérer l'arrivée de pièces dans l'installation. Il reçoit un (des) fichier(s) contenant l'ordre d'arrivée des produits dans l'unité et gère la répartition de ceux-ci sur les lignes en fonction la capacité de chaque ligne à traiter le produit et de leur charge instantanée. Dans le cas d'installation tournant en continu, ce module est capable de charger des fichiers différents en fonction de la période de la journée (par exemple pour simuler la fermeture de certains postes la nuit).
- **Les groupes** : ces objets sont utilisés pour décrire les groupes de cuves de même type, dans le cas de duplication de certaines ressources. Ils agrègent les différentes ressources des cuves constitutives, pour former une ressource à capacité non unitaire.

2.3.3. Réalisation de la bibliothèque et exemple de modèle

Pour chacun des objets de la bibliothèque, nous avons réalisé la programmation des règles (voir un exemple figure 39) ainsi qu'un dialogue de paramétrage permettant à l'utilisateur de saisir l'ensemble des paramètres de l'objet. Puis nous avons réalisé un modèle Siman/Arena décrivant le comportement l'objet. Dans notre modélisation, les entités représentent soit les produits physiques évoluant sur la ligne, soit les messages logiques de demande de réservation.

Le modèle d'une ligne est ensuite construit en plaçant dans celui-ci les objets nécessaires et en les paramétrant. Un exemple de modèle, réalisé à l'aide de cette bibliothèque, est présenté dans la figure 40.

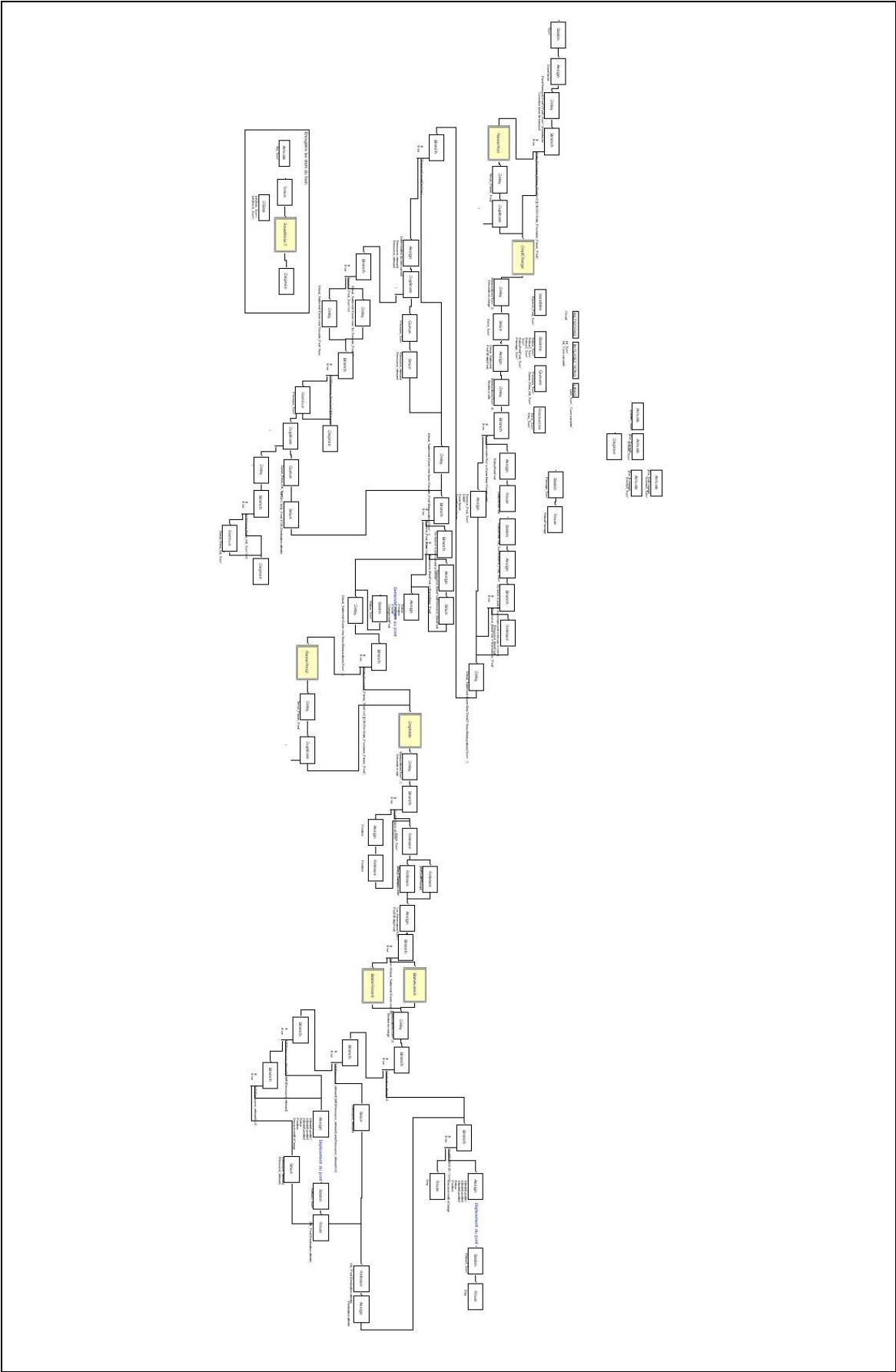


Figure 39 : Modèle logique du bain sous Siman/Arena

2.4. Exploitation des simulations

L'analyse des résultats de simulation est une phase importante du processus de développement d'un outil de pilotage. L'analyse doit permettre de comprendre le comportement émergeant afin d'affiner la définition des règles. Elle doit également fournir des indicateurs qualitatifs et quantitatifs pour évaluer les performances globales du système défini et mesurer l'impact des règles sur la performance.

Pour réaliser cette tâche, nous avons développé deux post-processeurs, permettant d'analyser la trace de l'exécution d'une simulation : l'un graphique, pour analyser le comportement émergeant, l'autre quantitatif, permettant une agrégation de la trace sous forme d'indicateurs de performances.

2.4.1. Analyse graphique des résultats

La **visualisation graphique** permet de comprendre le fonctionnement résultant du couplage des règles entre elles. Cette analyse est la principale source d'informations pour décrire et affiner les règles. C'est elle qui a permis de décrire et d'augmenter le nombre de règles (voire d'en supprimer) au fur et à mesure des essais successifs. Quelques unes de ces analyses ont été rapportées ci-dessous.

- **Les blocages (dead lock)** : les attentes servant à résoudre certains blocages, leur pénurie est la principale cause des blocages. Cette pénurie peut être due à un sous-dimensionnement des celles-ci ou à une sur-utilisation.
- **Les ressources goulots** : les nombreux dépassements sur une cuve sont dus, soit à une surcharge du pont alors indisponible pour sortir les pièces, soit à une indisponibilité de la cuve suivante en raison de sa surcharge.
- **L'utilisation des attentes** : la visualisation graphique permet d'évaluer la pertinence d'une décision de dégagement en attente. On remarque parfois l'inutilité d'un tel dégagement, qui aura pour conséquences d'augmenter la charge du pont par des mouvements chronophages. Cette analyse permet d'affiner les règles pour éviter de telles situations.

La figure 41 représente un exemple d'une simulation d'une portion d'installation industrielle B. On peut observer un dépassement dans un bain critique (bain 1). Ce dépassement est dû à une surcharge du pont qui n'a pas eu le temps de prendre la pièce plus tôt dans le bain 1.

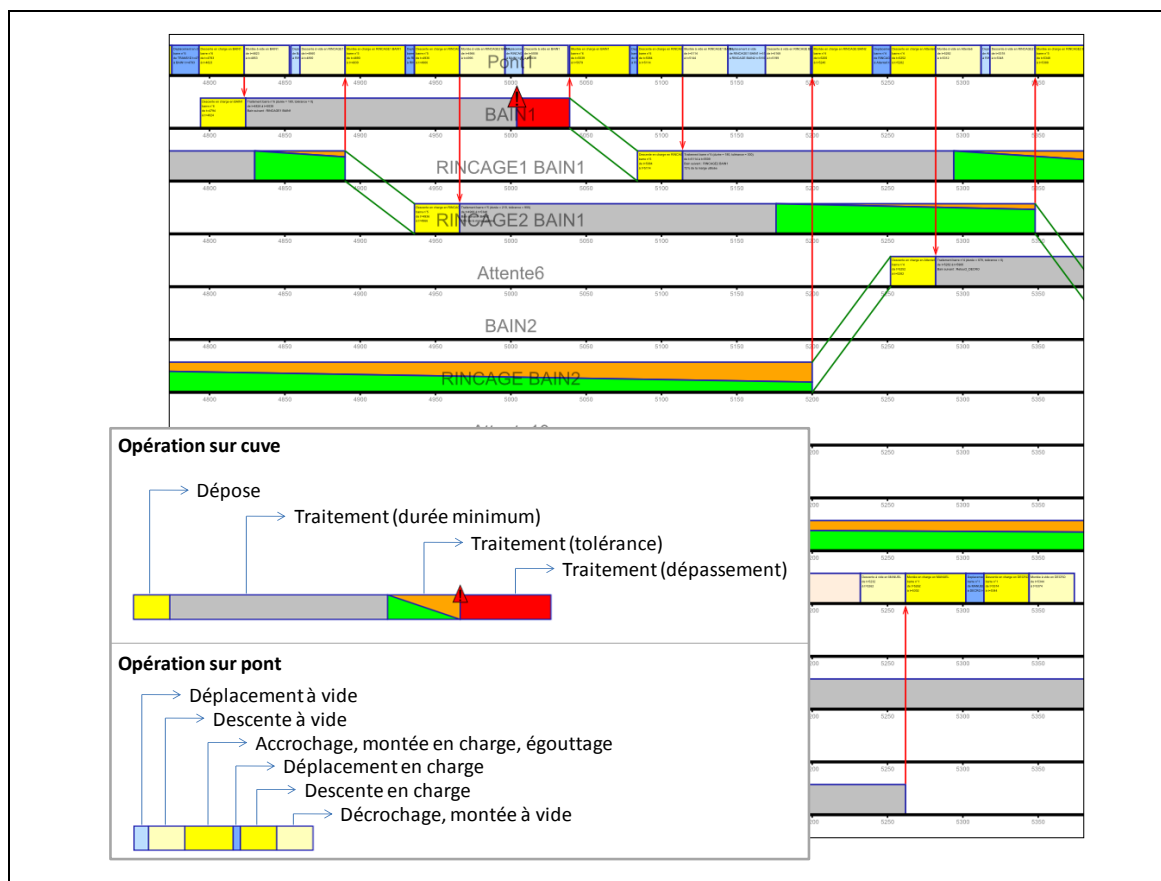


Figure 41 : Exemple de visualisation graphique

2.4.2. Indicateurs de performance

Les indicateurs quantitatifs permettent d'analyser l'évolution des performances de l'installation en fonction des différentes règles ajoutées, affinées, supprimées. Ils permettent de mesurer l'impact final sur la qualité des produits réalisés et les performances de l'installation. Les différentes analyses nous ont permis de sortir quelques indicateurs utiles. Nous les classeront ci-dessous en indiquant leur utilité dans le processus de définition des règles et d'implantation de l'unité :

- **Productivité**

- Nous utilisons comme indicateur de productivité le nombre de porteurs produits par quart (période continue de 8 heures). Cet indicateur est couramment utilisé dans l'industrie.

- **Utilisation du pont**

- **Taux d'utilisation.** Cet indicateur, égal à la durée d'activité du pont divisée par la durée de l'étude, renseigne l'utilisateur sur la charge du pont. Un pont peu chargé aura peu de chances de réaliser des missions en retard. Entre 0% et 50%, le pont n'effectue que très peu de missions en retard, entre 50% et 75%, le pont induit des retards sur les cuves non prioritaires, à partir de 75% le nombre de retards dans des cuves critiques devient trop important et génère trop de rebuts.
- **Taux d'attente.** Les attentes sont des intervalles de temps faisant partie de l'utilisation du pont. Elles correspondent au temps entre l'arrivée du pont au-dessus

d'une cuve et la date de fin de traitement de la barre. Elles sont provoquées par des durées d'anticipation d'appel trop longues. Une attente ne se produit que dans le cas où le pont n'a qu'une mission à effectuer. Ce taux tend vers 0% lorsque le pont a plusieurs missions en attente d'exécution. Ce taux permet de régler les durées d'anticipation, suffisamment élevé pour disposer du pont au bon moment, mais suffisamment faible pour ne pas lui faire perdre du temps d'attente.

- **Classification des missions.** Nous proposons de compter les missions par destination demandée (cuve, attente, navette, sortie) et par pont. Une répartition des missions par quart (8 heures de production) permet d'avoir une valeur normée de ces indicateurs. On remarque que le nombre global de missions par quart est limité par la capacité du pont. A partir d'une charge du pont de 75%, seule la répartition des missions change, leur somme reste stable. On observe alors une augmentation du nombre de missions vers les postes d'attente et une diminution des autres types de missions. Cette diminution traduit une baisse de productivité. La limitation du nombre de barres dans l'installation a pour effet de limiter la charge du pont, donc de limiter le nombre de dégagements vers des postes d'attente, et d'augmenter légèrement la productivité.
- **Nombre de missions par quart.** Le nombre d'opérations dans une gamme de traitement de surface peut varier de quelques opérations à une centaine d'opérations dans le cas de réalisation de circuits imprimés. Le nombre de missions par quart est alors un indicateur d'utilisation du moyen de production, indépendant de la longueur des gammes.

- **Utilisation des cuves**

- **Taux d'utilisation.** Le taux d'utilisation correspond à la durée réelle de trempe des produits divisée par la durée d'observation.
- **Taux de réservation.** Le taux d'utilisation comptabilise les durées entre le début de la réservation par la pièce située dans la cuve précédente jusqu'à la sortie de la pièce de la cuve. L'augmentation des enchainements obligatoires ou l'augmentation de l'anticipation a tendance à faire augmenter ce taux. Cet indicateur est utile pour choisir les cuves à dupliquer.
- **Nombre d'utilisation.** Le nombre d'utilisation d'une cuve correspond au nombre de pièces passées dans cette cuve. Cet indicateur permet de connaître le flux de pièces dans une cuve. Cet indicateur est une image du nombre de déplacements du pont vers cette cuve.

- **Dépassements**

- La qualité des produits réalisés par une installation de traitement de surface est liée au temps de dépassements de traitement. Il existe deux types de traitement : ceux où un dépassement aura peu d'influence et ceux où il conduira à la non qualité du produit. Le nombre de dépassements ainsi que la durée de ceux-ci et le bain concerné doivent être analysés finement pour chercher à éliminer tous les dépassements dans des bains critiques.

2.5. Analyse des résultats

Les résultats présentés dans cette partie sont issus de l'exploitation du modèle de simulation. Ce modèle a été configuré pour l'installation industrielle A. Les données de production configurées correspondent à des données réelles, extraites d'un an de production type. L'analyse des indicateurs permet d'observer l'évolution des performances de l'installation en fonction du paramétrage du modèle et de déduire certaines contraintes de configuration. Ces contraintes sont généralisables à toute installation correspondant à la classe de problèmes décrite au début de ce chapitre.

Le premier paramètre testé est la limite du nombre de barres gérées par ligne. La Figure 42 reprend l'évolution du temps de production (les unités ne sont pas importantes, seule l'évolution nous intéresse). Les instances non représentées correspondent à des instances conduisant à des blocages.

La première conclusion à tirer concerne la relation entre **le nombre de postes d'attente** et **le nombre maximal de barres par pont**. L'analyse des instances non représentées rapporte qu'au moment du blocage, tous les postes d'attentes d'au-moins une des lignes étaient remplis. Par conséquent, le nombre de postes d'attente doit être suffisamment grand pour ne pas provoquer de blocage. Un remplissage de tous les postes d'attente conduit au pilotage d'une installation sans attente avec les barres restantes. Dans ce cas, il ne doit rester qu'une seule barre dans la ligne pour éviter les blocages. La limite du nombre de barres par ligne doit être égale ou inférieure au nombre de postes d'attente plus 1. Cette limite peut alors être différente sur chaque ligne.

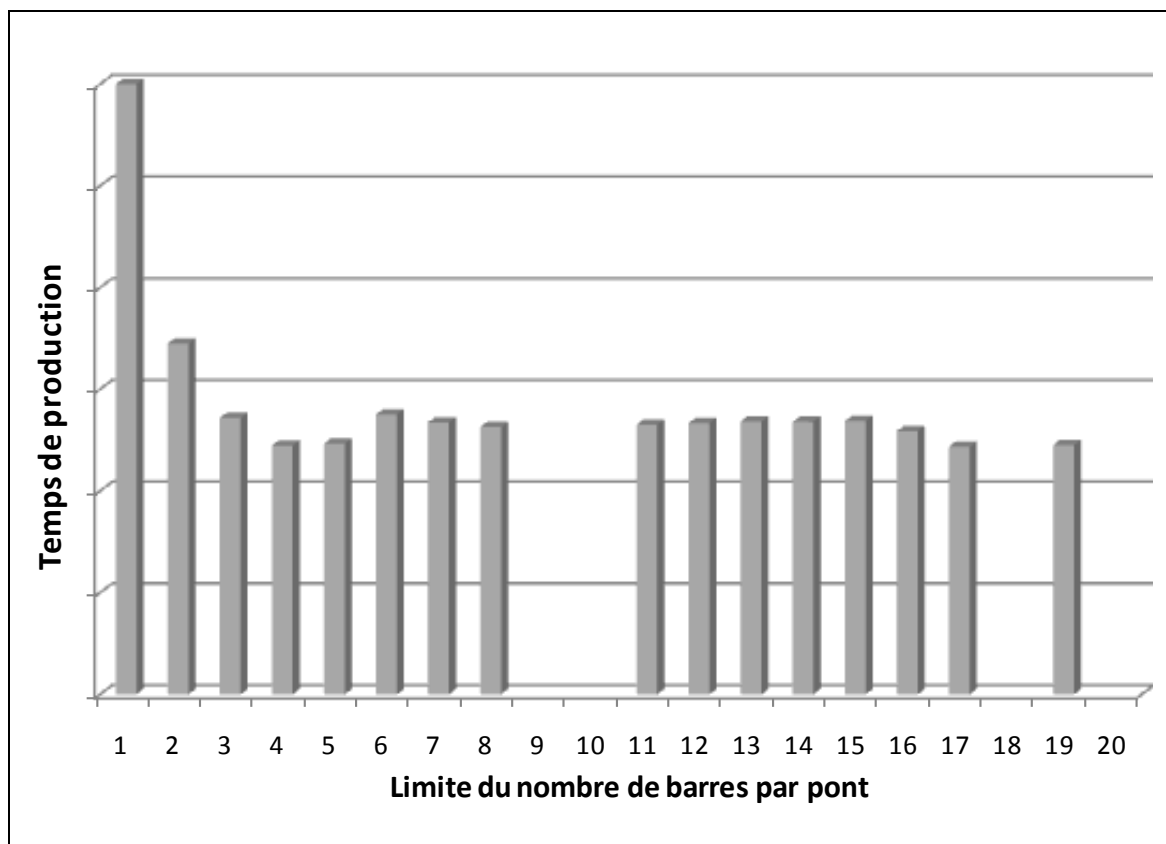


Figure 42 : Influence de la limite de barres sur le temps de production (4 attentes par ligne)

L'analyse de l'évolution du C_{max} (correspondant au temps utile pour produire une certaine quantité de produits) montre que l'augmentation de la limite au-delà d'un certain seuil a tendance à faire **augmenter le temps de production global**. En effet, à partir d'une certaine limite, les postes deviennent chargés et souvent indisponibles. Cette charge a pour conséquence la demande de dégagement en attente d'un nombre plus important de palonniers, produisant une augmentation de la charge du pont. Celui-ci étant une ressource goulot dans l'installation, l'augmentation de sa charge tend à faire diminuer la productivité. La volonté de charger l'installation aura alors un effet néfaste sur ses performances globales. Le dépassement d'une limite (ici 15 barres par ligne) conduit sur le long terme à une saturation de l'installation.

Une des contraintes du processus de traitement de surface est le non-dépassement de la durée de traitement dans les bains critiques. Lorsqu'on limite le nombre de barres par pont à 1, le pont est toujours disponible pour la pièce, on n'observe donc aucun dépassement de traitement. L'évolution du nombre de dépassements est reprise dans la figure 43. On remarque que plus la charge du pont augmente, plus le nombre de dépassements (à nombre de traitement équivalent) augmente. Il n'est pas possible, en pilotage purement réactif, de supprimer ces dépassements. Un pilotage réactif ne peut pas empêcher deux traitements critiques de se terminer en même temps, et donc de provoquer une indisponibilité du pont pour sortir une des deux pièces. La réduction du nombre de dépassements n'est que la conséquence de la saturation de tous les bains critiques provoquant ainsi l'attente des pièces sur des postes d'attente et non dans des bains critiques.

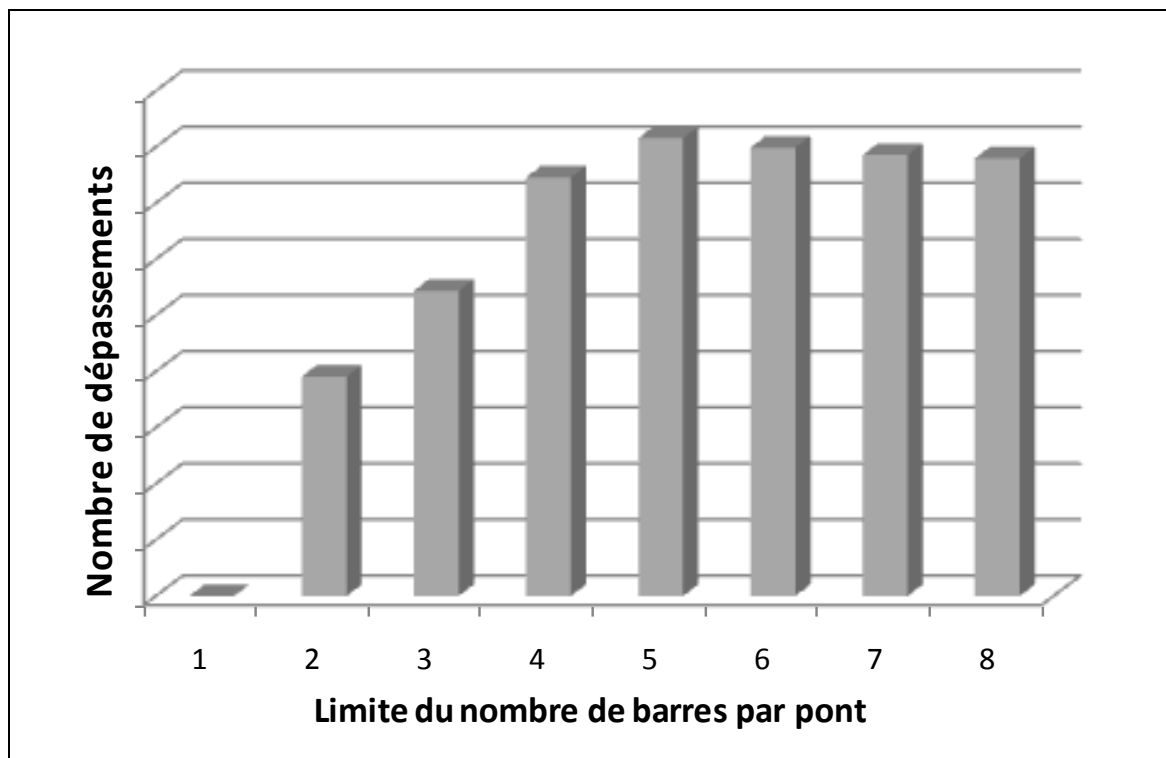


Figure 43 : Influence de la limite de barres sur le nombre de dépassements critiques

Ces deux analyses montrent le dilemme pour le réglage de la limite de capacité. La qualité des pièces impose de fixer la limite à 1, la productivité de l'installation (sans prendre en compte les rebuts)

aurait tendance à vouloir augmenter cette limite. En pilotage purement réactif, il n'existe pas d'autre possibilité que de dédier un pont à une barre.

La question du pilotage réactif des lignes de traitement de surface devient alors celle-ci : *existe-t-il un mode de pilotage dérivé du pilotage réactif, gardant ses avantages, permettant de produire plusieurs pièces, sans dépassement critique, en même temps ?* Les approches de résolution du RHSP dans la littérature se basent toutes sur des règles de priorité des missions affectées au pont, ces méthodes permettent de limiter les dépassements dans les cuves critiques, mais aucune méthode purement réactive ne peut bannir les dépassements critiques. Les travaux sur les RHSP ne parlent pas de ce problème ou tolèrent des dépassements dans les cuves critiques pour permettre un pilotage purement réactif.

3. Couplage entre pilotage réactif et prédictif

Il n'est pas envisageable, dans ce type de production, de tolérer des dépassements dans les cuves critiques. Cette contrainte n'étant pas possible avec un pilotage purement réactif, celui-ci est exclu. Nous présentons ici une méthode de pilotage basée sur le pilotage réactif (identique à celui présenté ci-avant) couplée à une méthode prévisionnelle. La couche prédictive aura pour objectif la planification des opérations de sortie des pièces dans des bains critiques. L'approche proposée conserve les avantages du pilotage réactif et y ajoute la sécurité d'un pilotage prédictif.

3.1. Bref état de l'art

On retrouve dans la littérature deux approches de couplage. La première approche présentée ci-dessous consiste à conserver un mode de pilotage réactif en y ajoutant une couche de contrôle. C'est une application de l'*Holon staff* de l'architecture holonique de référence PROSA présentée dans [Van Brussel H. *et al.*, 1998]. Cette méthode se base sur un fonctionnement réactif contrôlé par une entité supérieure qui peut ou non modifier les cours de l'exécution du système. Dans notre cas cette entité (ou Holon staff) aura un planning prédictif de certaines tâches qui lui permettra d'intervenir ou non dans le processus de choix de mission du pont.

La seconde approche consiste à établir un pilotage prédictif (ou dynamique) et à y introduire une couche réactive en cas d'occurrence d'un événement imprévu. Ce mode de pilotage présenté dans [Sun J. *et al.*, 2001] permet d'avoir une vision des actions à venir tout en conservant la continuité du pilotage. Dans l'article cité, les auteurs utilisent le pilotage réactif pour prendre en compte rapidement des changements de commandes clients dans la production, en attendant la prise en compte des modifications dans le pilotage prédictif.

3.2. Principe du couplage

Nous remarquons que les seuls problèmes non résolus par un pilotage réactif concernent la sortie des barres contenues dans des bains critiques lorsque deux traitements se terminent en même

temps. La couche prédictive aura alors pour mission de réserver des plages futures d'utilisation pour la sortie des pièces. Une mission du pont pourra être exécutée uniquement dans le cas où elle ne rentre pas en conflit avec une plage déjà réservée. De plus, un déplacement vers un bain critique ne pourra se faire que dans le cas où une plage de sortie de ce bain est insérable dans le planning prédictif du pont.

Le principe général du pilotage réactif présenté précédemment reste valable ; toutes les règles de celui-ci sont conservées. Les bains arrivant en fin de traitement cherchent à réserver le bain suivant puis effectuent au pont une demande de déplacement. La modification apportée apparaît dans le processus de sélection de mission intégré à la logique de fonctionnement du pont. Ce nouveau processus est décrit dans la figure 44.

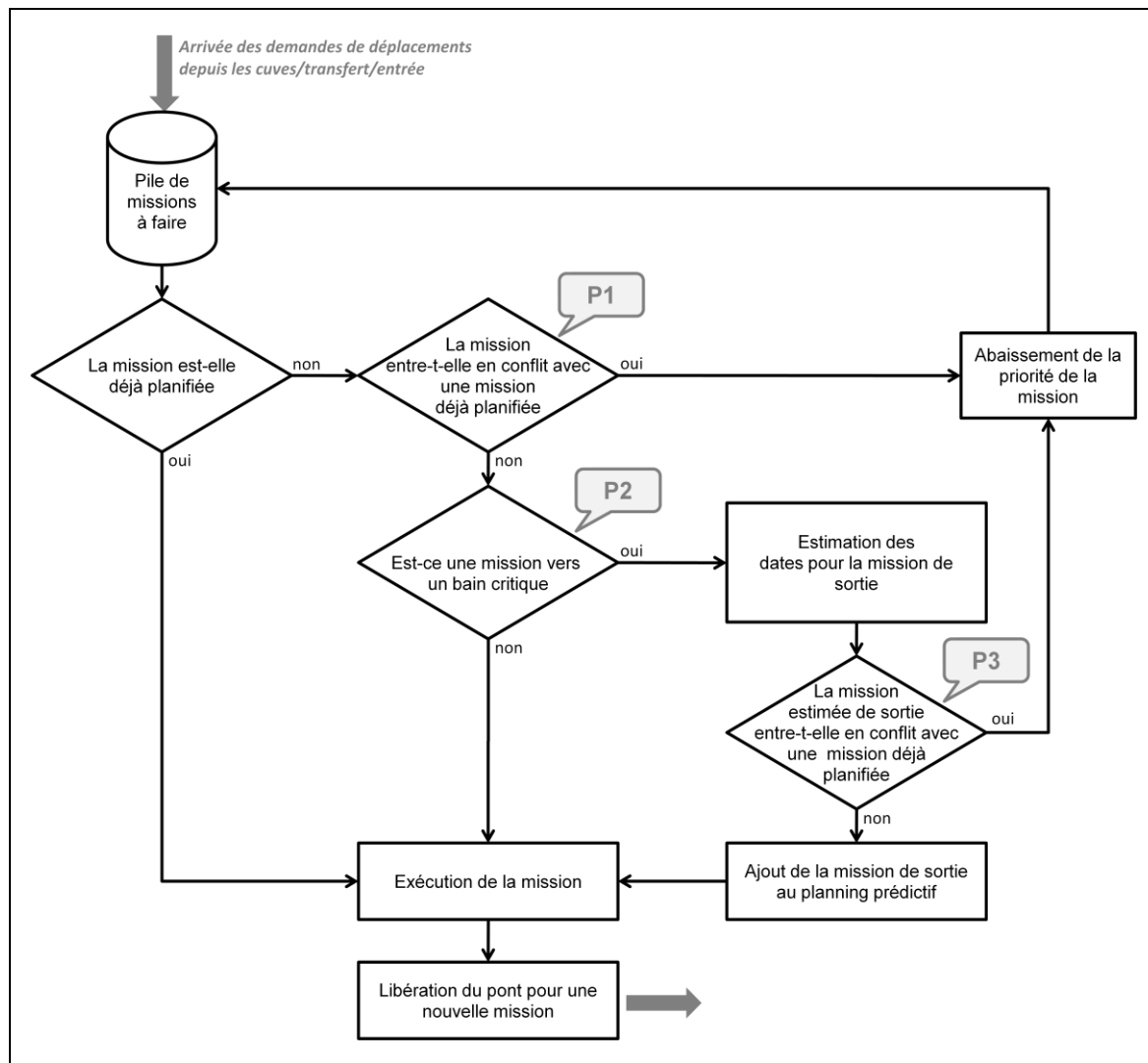


Figure 44 : Couche prédictive de la sélection des missions du pont

La couche prédictive apporte trois nouvelles règles. Les règles planifiées dans le planning prédictif ne suivent pas ces trois règles et sont directement exécutées quel que soit l'état du planning. La **règle P1** impose au pont de n'exécuter que des missions qui ne vont pas rentrer en conflit avec le planning prédictif établi. Dans un cas d'exécution sans aléa, cette règle assure la disponibilité du pont pour une opération planifiée. La **règle P2** différencie les missions vers des cuves critiques. Les missions ne

rentrant pas en conflit avec les plages réservées et à destination d'une cuve non critique seront exécutées. Dans le cas de missions vers une cuve critique, la **règle P3** vérifie la disponibilité du pont à la fin du traitement critique, et ajoute la mission de sortie au planning prédictif. Cette règle peut être appliquée en chaîne dans le cas de plusieurs bains successifs critiques.

3.3. Modèle holonique

L'architecture holonique de référence PROSA est particulièrement bien adaptée à la modélisation du pilotage réactif du traitement de surface.

Dans le mode de pilotage purement réactif présenté au début de ce chapitre, la modélisation holonique du problème se base uniquement sur les Holons produits, ressources et ordres. Le **Holon produit** est l'image du porteur de pièces qui va effectuer une succession de traitement dans l'installation. Le *Holon produit* sera en relation avec un **Holon ordre** qui dispose des informations de production (durée, enchaînement, contraintes). La coordination d'un *Holon produit* et d'un *Holon ordre* sera capable de définir l'ordre des opérations du produit, donc l'enchaînement des besoins des différentes ressources. Cette connaissance de la production leur permettra d'interroger les **Holons ressource** (transporteur, cuve, navette) pour évaluer leur capacité à répondre à leur besoin. La gestion de la zone commune est aussi modélisée par un *Holon ressource*, lié aux différents transporteurs. La représentation graphique de cette modélisation est représentée dans la figure 45.

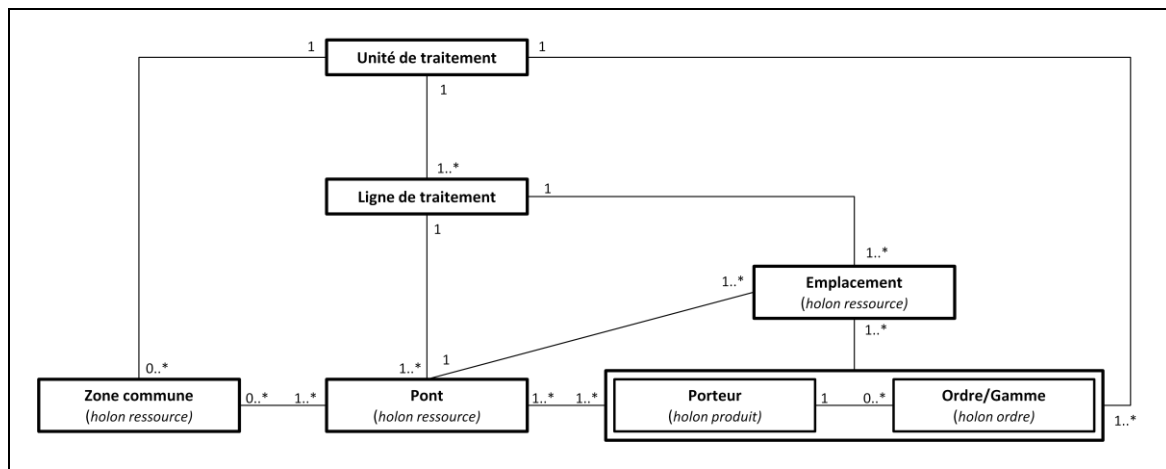


Figure 45 : Diagramme UML du pilotage réactif

Le couplage entre le modèle de pilotage réactif présenté ici et le pilotage prédictif se modélise par l'insertion d'un **Holon staff** dans l'architecture de la figure 45. Chaque pont sera muni d'un *Holon staff* qui enregistre les plages de réservation du pont auquel il est associé. Lorsque le pont souhaite effectuer un déplacement, il interroge le *Holon staff* qui lui est associé. Celui-ci lui donne une indication sur la possibilité d'un conflit avec une plage réservée. En cas de non réponse du *Holon staff*, le *Holon ressource pont* est capable de prendre des décisions (peut-être non-sûres). Ces décisions correspondront au mode de pilotage réactif. Le *Holon staff* n'a donc, conformément aux spécifications de l'architecture PROSA, qu'un rôle de conseil auprès des autres Holons.

3.4. Application à un cas industriel

Ces règles ont nécessité, dans la bibliothèque *Siman/Arena*, une petite adaptation des différents objets de position (cuve, attente) et des gammes pour ajouter quelques réglages. De plus, elles ont conduit à une refonte complète du programme de gestion du pont. La file d'attente ordonnée du pont (utilisée dans un pilotage purement réactif) a été remplacée par un algorithme réalisé à partir du diagramme de la figure 44, mettant en jeu des algorithmes plus compliqués d'interrogation des disponibilités temporelles des baignoires. Le planning prédictif a, quant à lui, été modélisé par une file d'attente contenant des éléments caractérisés par une date de début et une date de fin de réservation. Ces éléments sont supprimés de cette file dès que leur date de début est atteinte (pour libérer le pont et les ressources associées en cas d'aléa).

Le principal objectif de ces nouvelles règles est de supprimer les dépassements sur les durées de traitement dans les cuves critiques. On observe que les dépassements obtenus avec un pilotage purement réactif (visibles dans la figure 43) ont complètement disparus.

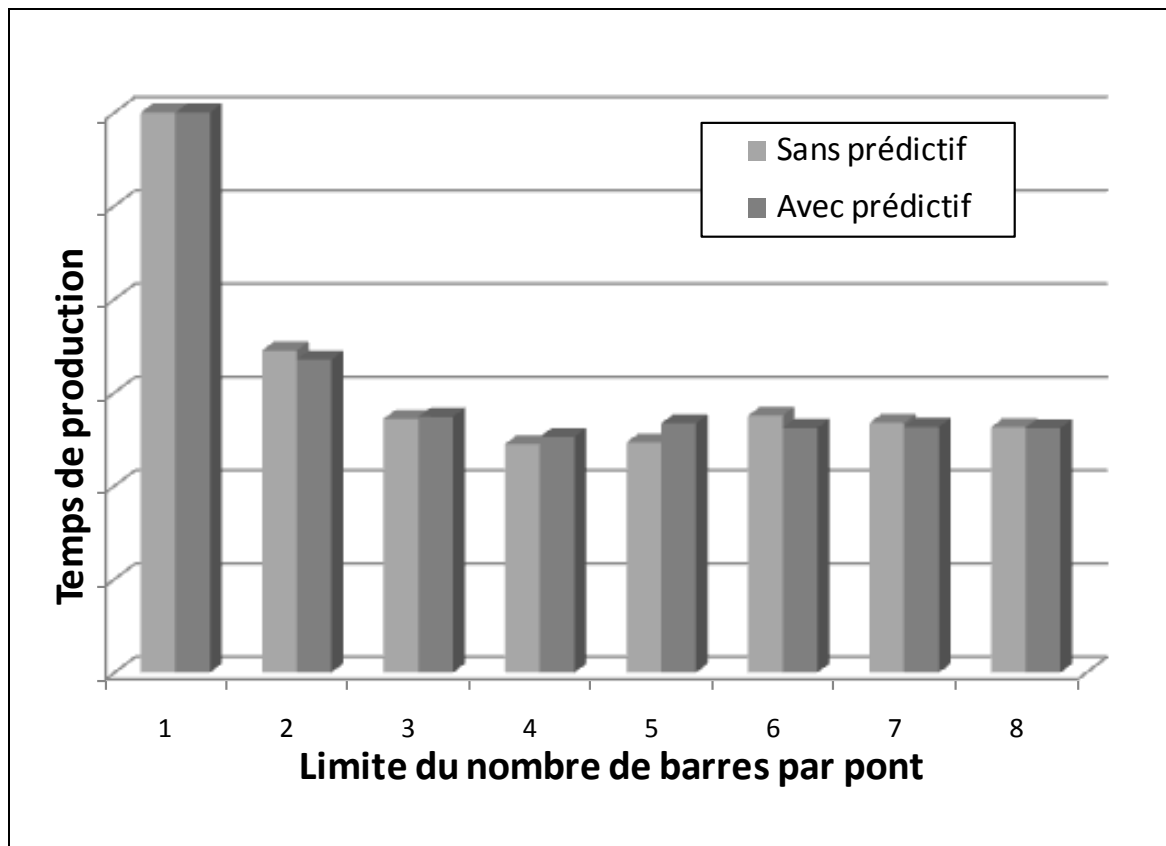


Figure 46 : Évolution des performances avec le planning prédictif

D'un autre côté, la comparaison avec les instances utilisées pour évaluer le pilotage purement réactif montre une faible modification des performances quantitatives. La comparaison représentée dans la figure 46 montre que les changements de performances sont minimes (moins de 5%) et pas toujours au profit du pilotage purement réactif.

4. Implémentation

4.1. Choix d'un outil d'ordonnancement

Le choix d'un outil de pilotage (réactif ou dynamique) doit être validé avant la mise en production de celui-ci. Pour ce faire, nous avons développé une plateforme de test des outils basée sur un émulateur, et un analyseur de la trace du système (basé en grande partie sur celui exposé dans la figure 41 page 72).

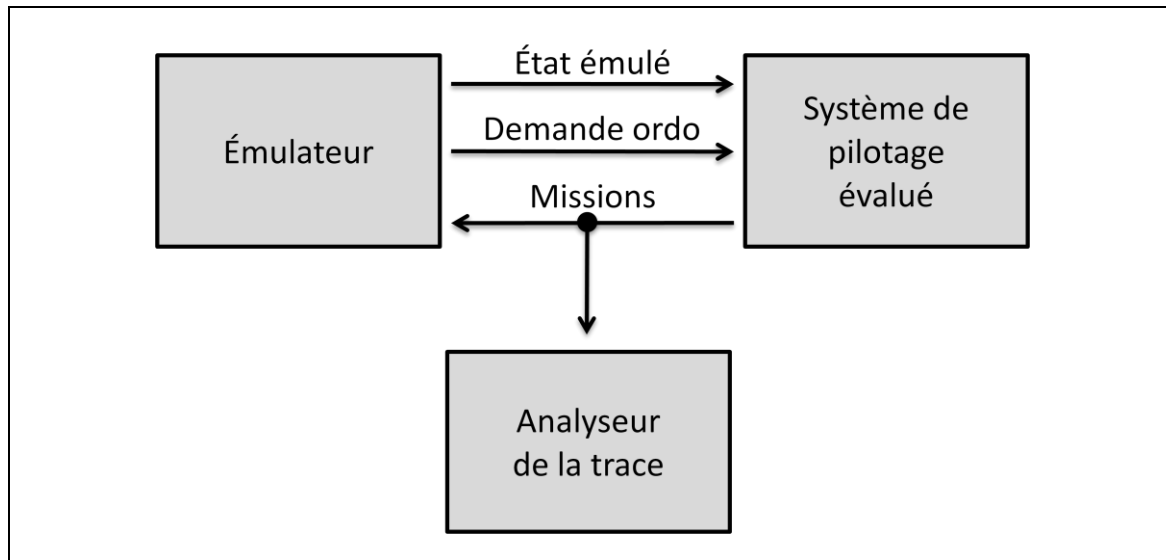


Figure 47 : Évaluation d'un système de pilotage

Nous avons mis à l'épreuve quatre systèmes de pilotage dans ce système d'évaluation. Trois logiciels réputés du commerce, fonctionnant sur un ordonnancement prédictif, et un système développé pour l'occasion, fonctionnant de façon réactive.

La première difficulté dans l'utilisation des logiciels du commerce réside dans l'implémentation des règles d'ordonnancement. Les deux premiers logiciels d'ordonnancement, trop peu personnalisables n'ont pas permis d'intégrer toutes les règles de pilotage. Ils ont donc été abandonnés pour cette étude.

Le troisième logiciel d'ordonnancement (*Preactor*) a pu intégrer toutes les règles et prendre en compte l'état courant du système. Ainsi, l'émulation (et plus tard, la supervision réelle) envoie à ce logiciel l'état du système et effectue une demande d'ordonnancement. Le logiciel d'ordonnancement applique les règles et renvoie au système (émulateur ou supervision) l'ensemble des missions à venir. Ces missions sont exécutées jusqu'à ce que la dérive temporelle induite par les aléas soit trop grande. Dans ce cas, un nouvel ordonnancement est demandé, à partir du nouvel état du système. Ce principe a l'avantage de donner un aperçu des prochaines opérations, mais bloque la production pendant l'ordonnancement (environ 1 minute, fonction de la charge à ordonnancer). Pour cette implémentation, *Preactor* a été configuré en mode périodique (par opposition au mode événementiel), et les règles ont été implantées directement à l'aide du langage natif de *Preactor* (utilisant la technologie *.net*) par un ingénieur de la société.

Le système réactif a, quant à lui, un temps de réponse d'un dixième de seconde, donc négligeable devant l'échelle temporelle du problème. Cette solution offre le même ordonnancement final dans le cas d'un système piloté sans aléa, et présente une meilleure prise en compte de ceux-ci dans le cas d'apparitions d'aléas.

4.2. Pilotage réactif ou pilotage dynamique ?

L'implémentation d'un ordonnancement réactif dans une installation industrielle peut prendre plusieurs formes. La première implémentation possible consiste à déployer les règles dans l'organe de pilotage (l'automate ou la supervision d'atelier). Quel que soit l'organe pilotant, il n'effectue aucune prévision mais applique les règles définies sur occurrence d'un événement. Cette implémentation a l'avantage de conserver au mieux les intérêts d'un pilotage réactif, mais garde l'inconvénient d'une absence de vision sur les prochains mouvements du système. Dans un atelier où l'humain n'intervient pas, la vision des prochains événements n'a que peu d'importances. Dans d'autres cas (préparation des opérations manuelles, préparation de l'expédition...) il peut être souhaitable de connaître les prochaines opérations réalisées. Dans ce deuxième type d'implantation, l'appel à un module d'ordonnancement appliquant les règles est nécessaire. Ce module, décrit dans [Cardin O., 2007], aura pour mission d'extraire l'état du système pour le charger dans un simulateur et d'effectuer une simulation. La trace de la simulation sera alors l'ordonnancement dynamique proposé à la supervision. Dans ce cadre d'un ordonnancement dynamique, l'occurrence d'un aléa modifiant le planning établi provoquera la reconstruction de celui-ci à partir du nouvel état réel du système. Ces deux modes d'implémentations sont schématisés dans la figure 48.

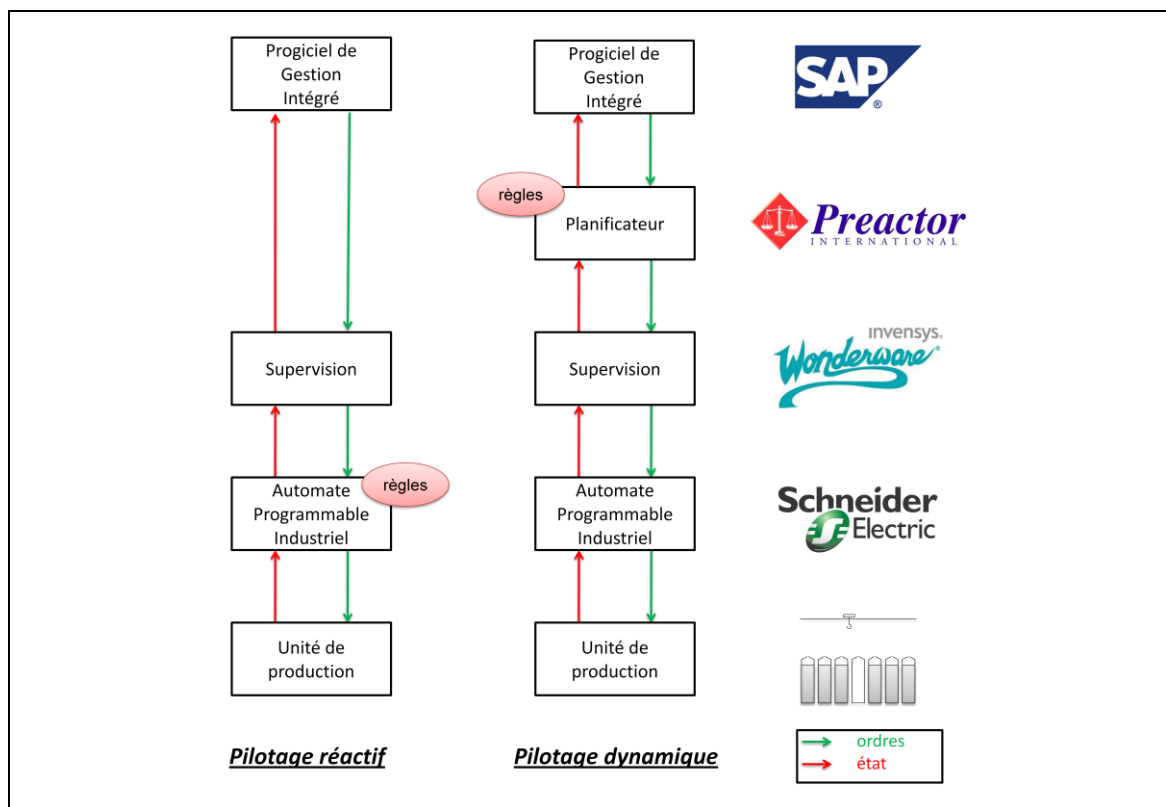


Figure 48 : Exemple d'implémentation dynamique et réactive

Dans le contexte industriel dans lequel ces travaux ont été effectués, il a été décidé de conserver une part de prévision dans le système de pilotage afin de laisser au pilote humain une certaine vision de l'environnement de pilotage. Cette vision lui permet d'anticiper la préparation de certaines opérations. Ce choix a été fait au détriment de la réactivité. En effet, l'occurrence d'un aléa aura pour effet la remontée d'une demande de planification jusqu'au planificateur, l'établissement d'un nouveau planning et la redescende de celui-ci jusqu'à la production. Une des tâches complexes, que nous n'aborderons pas ici, consiste à conserver un pilotage cohérent de la production dans le cas où le planning établi devenait inapplicable en raison de l'occurrence d'une perturbation. Le pilotage cohérent souhaité doit chercher à maintenir la qualité de la production, notamment éviter les dépassements de durée des traitements. De plus, les décisions du pilotage transitoire ne doivent pas rentrer en conflit avec le futur planning proposé par le moteur de planification.

Pour comparer les stratégies dynamique et réactive, nous avons développé un modèle de simulation implémentant les différentes règles de pilotage. A intervalle régulier, nous préemptons les ponts pendant une durée donnée. Cette préemption correspond au temps d'ordonnancement. Dans le cas d'un pilotage dynamique, la fréquence de ré-ordonnancement sera faible, mais la durée relativement longue. Dans le cas du pilotage réactif, les préemptions arriveront plus souvent mais seront de durée plus courte. Les résultats sont présentés dans le tableau 9. Les essais ont été réalisés pour une production aléatoire identique pour tous les tests.

| Essai | Intervalle entre 2 ordonnancements | Durée d'un ordonnancement | Nombre de missions critiques | |
|-----------|------------------------------------|---------------------------|------------------------------|-----------|
| | | | A l'heure | En retard |
| Dynamique | 9 à 11 min. | 50 à 70 secs | 1362 | 138 |
| | 19 à 21 min. | 50 à 70 secs | 1431 | 69 |
| | 29 à 31 min. | 50 à 70 secs | 1443 | 57 |
| Réactif | 50 à 70 secs | 1 à 2 secs | 1500 | 0 |

Tableau 9 : Comparaison des dépassements en fonction de la stratégie de ré-ordonnancement

Ces résultats montrent que le pilotage réactif, dans ce cas du traitement de surface, est le mode de pilotage qui permet de ne pas avoir de dépassement dans les traitements critiques. Le pilotage prédictif étant encore plus rigide que le pilotage dynamique, ses performances ne sont pas évaluées et une comparaison des résultats obtenus ici avec des algorithmes existant n'aurait que peu de valeur : le pilotage prédictif ne prend pas en compte toutes les contraintes du problème, notamment l'arrivée de nouveaux produits et la survenue d'aléas.

4.3. Solution adoptée

Sans rentrer dans les détails du choix industriel, une solution de pilotage dynamique basé sur *Preactor* a été adoptée pour les raisons suivantes :

- Solution réactive plus souple que la solution actuelle
- Prise en compte de la plupart des contraintes
- Possibilité d'optimisation dans les prochaines versions
- Diffusion du planning de chaque bain pour planifier les opérations de maintenance et les opérations manuelles

Notons par contre que l'outil PREACTOR construit bien son ordonnancement par simulation, en appliquant l'ensemble des règles dynamiques qui ont été présentées dans cette thèse.

Cette solution présente l'inconvénient d'introduire des dépassements (dans certains traitements critique, lorsque des aléas interviennent pendant une re-planification) évitables par l'adoption du couplage prédictif/réactif. Une solution hybride, basée pour le pilotage sur un pilotage réactif, et pour l'affichage sur le pilotage dynamique, permettra de combiner les avantages des deux solutions.

5. Conclusion

Dans ce chapitre, nous avons présenté un exemple de pilotage réactif des unités de traitement de surface. L'analyse des résultats de simulation sur deux cas industriels ont permis de mettre en évidence les lacunes du pilotage réactif dans ce problème à fortes contraintes temporelles, et l'impossibilité de satisfaire les contraintes de durée maximale de traitement dans les cuves critiques. Ces manquements conduisent à l'impossibilité d'appliquer cette méthode au milieu industriel, et supposent donc la recherche d'autres solutions. Le couplage entre le pilotage réactif initial et un pilotage prédictif a été présenté pour palier aux dépassements. L'implémentation de ce couplage en simulation a conduit à la validation par le secteur industriel, du concept scientifique développé. Enfin, la prise en compte de la dimension humaine de l'entreprise a conduit à la transformation du *pilotage réactif* en *pilotage dynamique* afin de présenter aux opérateurs les futurs mouvements et opérations du système.

Chapitre 4 : Optimisation de la topologie d'une unité de traitement de surface

| | |
|--|-----------|
| 1. La problématique | 86 |
| 1.1. L'implantation dans le cycle de vie de l'atelier | 86 |
| 1.2. Le champ des solutions d'implantation | 86 |
| 2. Quelques méthodes d'implantation | 88 |
| 3. Méthodologie de résolution : algorithme génétique..... | 89 |
| 3.1. Création d'une population initiale | 90 |
| 3.2. Mécanisme de reproduction/sélection | 91 |
| 3.3. Mécanisme de croisement..... | 92 |
| 3.4. Mécanisme de mutation | 93 |
| 3.5. Mécanisme de réparation..... | 94 |
| 3.6. Test d'arrêt..... | 94 |
| 4. Application à un cas industriel | 94 |
| 4.1. Présentation du cas industriel | 94 |
| 4.2. Exécution de l'algorithme | 95 |
| 4.3. Résultats..... | 96 |
| 5. Conclusion | 98 |

Dans le chapitre précédent nous avons proposé une méthode de pilotage d'une unité de traitement de surface. Cette méthode prenait comme acquis l'implantation de l'atelier (position relative des différentes ressources). Ce chapitre élargit le problème du traitement de surface à la modification de l'implantation de l'installation dans l'optique d'optimiser ses performances. Dans le cadre d'une implantation initiale ou d'une réimplantation, l'optimisation est un point stratégique qui vise à améliorer la productivité de l'installation. La recherche d'optimalité, ou du moins la recherche d'une amélioration de l'implantation, consiste à définir une disposition qui permette d'augmenter la productivité. Dans les installations de traitement de surface, les ressources critiques étant les ponts, l'optimisation de la topologie aura pour effet de réduire les temps de déplacement de ceux-ci en regroupant les cuves par gamme.

Nous commencerons cette étude par la présentation du problème d'implantation et les contraintes que nous lui avons imposées. Nous établirons le champ de solutions acceptables puis nous proposerons une méthode *itérative* permettant une recherche d'optimalité dans ce champ des implantations possibles. La méthode proposée est une généralisation des travaux présentés dans [Chové E. *et al.*, 2010] basée sur un algorithme génétique. Nous terminerons cette étude par quelques résultats sur l'implantation du problème industriel B.

1. La problématique

1.1. L'implantation dans le cycle de vie de l'atelier

L'étude proposée dans ce chapitre peut être appliquée, soit lors de la conception d'un atelier de traitement de surface, soit pendant une phase de réimplantation. La réimplantation d'un tel atelier demandant de gros moyens et imposant la vidange et la manipulation des cuves, celle-ci ne peut être effectuée que rarement. Cette contrainte impose, lors de l'étude, d'avoir une bonne vision de la charge à produire. En effet, une implantation optimale est liée à un scénario donné. Cette étude doit donc être précédée d'une étude de marché qui établit les différents scénarios qui se dérouleront pendant la vie de l'installation. Ce n'est qu'une fois le plan de charge défini, qu'il est possible d'optimiser l'implantation.

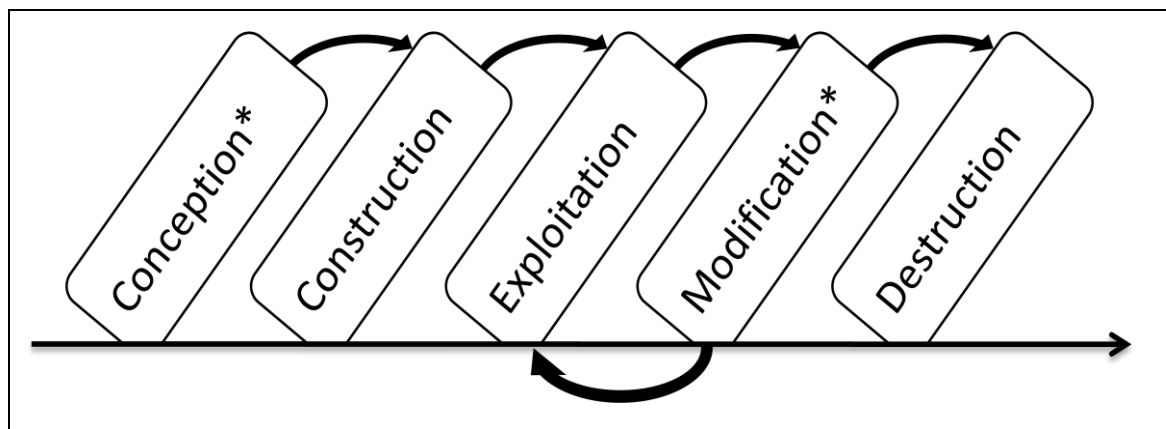


Figure 49 : Phases d'implantation dans le cycle de vie d'un atelier de production

Dans la suite de ce chapitre, nous ne nous intéresserons pas au moment de la réimplantation ni au coût engendré. L'objectif consiste à optimiser le moyen et évaluer le gain en performances en fonction d'un plan de charge donné.

1.2. Le champ des solutions d'implantation

Le problème que nous traitons concerne l'implantation des cuves. Nous n'aborderons pas dans ce chapitre le problème de dimensionnement de l'installation (nombre de cuves, nombre de ponts, dimensions des postes d'entrée/sortie). L'implantation, telle que nous l'étudions, concerne la position relative des différents postes les uns par rapport aux autres.

On distingue deux types de postes : les postes à deux connexions (cuve, attente, chargement, déchargement) et les postes à quatre connexions (transfert). Ces deux types sont repris dans la figure 50.

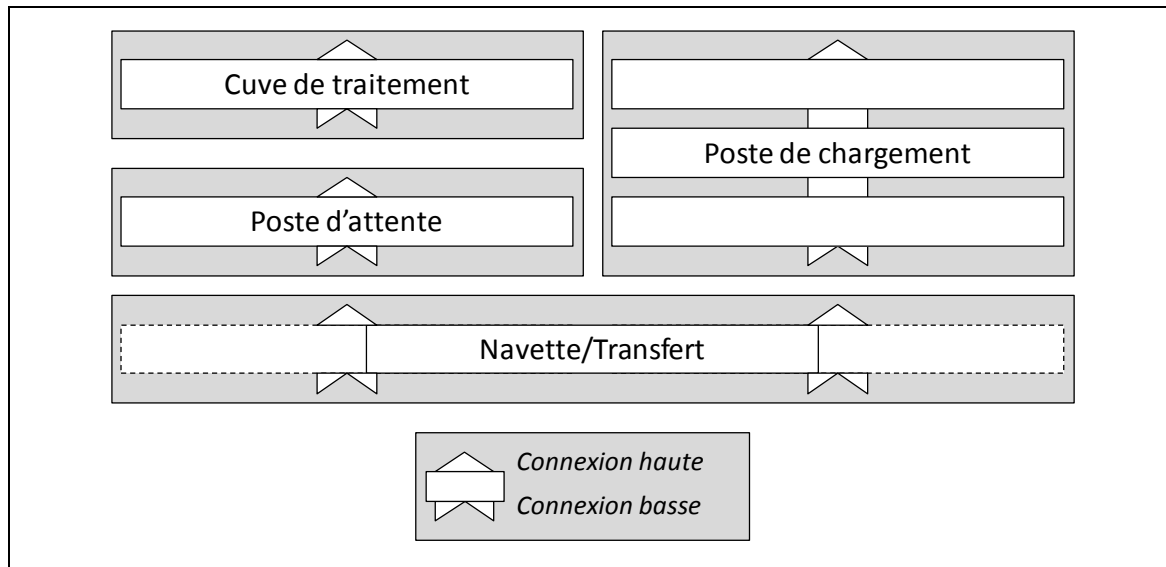


Figure 50 : Typologie des emplacements à implanter

L'objectif de l'étude présentée ici est de trouver un 'assemblage' intelligent des connexions de type *haute* et de type *basse* pour permettre à l'installation créée de produire efficacement les pièces. Un problème à n_h connexions haute et n_b connexions basse (sans contrainte particulière) peut avoir jusqu'à $(\min(n_h, n_b)!)^2$ topologies possible. Par exemple, l'implantation industrielle B (installation de taille moyenne) présentée dans le chapitre 3 contient deux navettes, 3 blocs d'entrée/sortie, quatre attentes, seize cuves de traitement. L'analyse exhaustive des implantations mènera à l'étude de plus de 10^{28} topologies différentes.

Toutes les topologies ne sont pas possibles : certaines contraintes inhérentes au problème particulier peuvent réduire le champ des solutions. Par exemple, la contrainte de placer un bloc en début de ligne supprime une connexion basse ; la contrainte de mettre un poste particulier en fin de ligne supprime une connexion haute ; la contrainte de lier deux blocs supprime une connexion haute et une connexion basse ; la séparation des cuves par ligne ajoute à chaque liaison une contrainte d'appartenance à la même ligne. Il est possible de multiplier les contraintes, cependant, le problème restera de complexité exponentielle en fonction du nombre de blocs. L'exploration exhaustive du champ des topologies possibles n'est pas envisageable pour des problèmes de grande envergure.

Il existe autant de problèmes que d'installations de traitement de surface. Les contraintes imposées sur le champ des solutions peuvent le réduire pour des problèmes de disposition d'atelier, d'interfaces avec les autres ateliers, ou d'autres contraintes spécifiques. L'étude que nous ferons présentera quelques unes de ces contraintes, sans en assurer l'exhaustivité.

2. Quelques méthodes d'implantation

Il y a deux grandes étapes dans les méthodes classiques de définition de **topologies d'atelier**. La première consiste à répartir les moyens en îlots les plus autonomes possibles. La seconde consiste, pour un îlot donné, à trouver la meilleure implantation possible. Il existe de nombreux travaux dans la première famille de méthode, par exemple les auteurs de [Vin E. *et al.*, 2010] proposent une méthode de répartition basée sur un algorithme génétique. Le problème de formation de cellule est important pour les installations de traitement de surface multi-lignes : quels bains doit-on placer sur chacune des lignes pour avoir le moins d'échanges entre les lignes ? Les méthodes présentes dans la littérature pour implanter les machines au sein d'un îlot de fabrication s'intéressent généralement au coût de transport de pièces, sans prendre en compte la ressource de transport, et ses déplacements à vide. Le problème du pilotage des transporteurs est lié autant à l'ordre des opérations dans une gamme (déplacement en charge) qu'à l'ordre de traitement des opérations entre les pièces (déplacement à vide). Dans le cas des unités de traitement de surface où la ressource critique est l'organe de transport, ces méthodes ne sont pas applicables directement.

On retrouve également, dans le **domaine de l'électronique** ou dans le **domaine de la télécommunication**, quelques travaux sur l'implantation des composants sur un circuit imprimé. Ces travaux visent à réduire l'encombrement d'un montage électronique. L'optimisation passe alors par la recherche de réduction des longueurs de canaux de communication pour réduire l'encombrement global. [Scheffel M., 2005] propose une méthode d'optimisation basée sur une évaluation négative de chaque topologie. L'auteur affecte un coût à chaque ressource, à chaque lien entre les ressources, et à chaque trafic sur un lien. La somme de ces coûts renseigne sur la qualité d'une topologie traversée par un flux de données. L'objectif est alors de trouver la topologie présentant le plus faible coût. La combinaison des trois coûts permet à l'algorithme de déterminer à la fois la dimension de l'installation (nombre de ressources) et leur organisation. Le problème mathématique a été résolu par programmation linéaire sous *CPLEX*⁸ sur un cas de réseau à 14 nœuds. Le résultat montre la nécessité d'un réseau maillé de forte densité. Il est possible d'appliquer cette méthode au problème d'implantation d'atelier en considérant chaque machine comme un nœud du réseau, chaque trajet comme un lien à capacité finie en fonction de la dimension des ressources de transport. Cependant, le modèle proposé se base sur la disponibilité d'une ressource d'un lien dès que le transfert est terminé. Cette contrainte imposerait d'ignorer les temps de déplacement à vide pour les ressources de transport. Dans le cadre du traitement de surface, les ressources de transport étant les ressources critiques, cette méthode est inapplicable.

Dans le domaine de la définition des **réseaux de commerces**, certains travaux se sont attachés à la recherche de la position optimale des fournisseurs, des clients ou des dépôts de véhicules pour minimiser les transports. Ces travaux ont le même objectif que celui de la définition d'une implantation d'atelier dont les ressources de transport sont critiques. Dans [Wasner M. *et al.*, 2004], les auteurs proposent une heuristique pour la définition de la localisation des dépôts et des hubs

⁸ Solution logicielle développée par la société ILOG

pour le problème des tournées de véhicules. L'heuristique permet de chercher le nombre de dépôts et leur position au sein d'un réseau routier. L'heuristique se base sur une analyse successive de différentes solutions (méthode évolutionniste). Pour chaque solution, une tournée de véhicules est calculée afin de déterminer les performances de l'implantation. Cependant ce problème diverge du problème du traitement de surface en raison de la nécessité de la préexistence d'un réseau de fournisseur et de clients.

3. Méthodologie de résolution : algorithme génétique

Les méthodes de résolutions présentes dans la littérature ne permettent pas de prendre en compte à la fois le problème de localisation des ressources et celui de la criticité des ponts. Ce chapitre propose une nouvelle méthode d'exploration du champ des implantations visant à trouver une implantation qui maximise la productivité (ou tout autre objectif).

L'évaluation de chaque topologie au cours de l'algorithme itératif sera faite par simulation à événements discrets. On retrouve, dans la littérature, de nombreuses utilisations de la **simulation à événements discrets** pour l'optimisation. Les travaux se répartissent en trois groupes correspondant aux trois types d'optimisations d'un moyen de production, exposés dans la conclusion du chapitre 2 (page 83). Dans [Huyet A.-L. *et al.*, 2004], les auteurs proposent une méthode d'optimisation évolutionniste par apprentissage et une évaluation des solutions par simulation. La méthode proposée optimise l'ordonnancement des opérations, et a été appliquée dans le cas d'un job-shop à cinq machines. Plus généralement, l'auteur de [C Fu M., 1992] propose un algorithme pour optimiser tout système de pilotage. L'algorithme génétique exposé agit sur un nombre fini de paramètres continus pour trouver un réglage adapté. Tout comme dans notre étude, les performances de chaque réglage sont évaluées par simulation.

Les travaux présentés ici ne s'intéressent pas à la définition du système de pilotage, mais à l'élaboration du système de production. Il nous est paru intéressant de prendre un algorithme évolutionniste permettant de croiser entre elles plusieurs topologies afin d'élaborer de nouvelles propositions. C'est pour cela que nous avons choisi un algorithme génétique, qui permet de combiner plusieurs solutions pour en faire ressortir des caractéristiques communes. Les caractéristiques intéressantes des solutions d'une génération sont reproduites dans la génération suivante.

La difficulté de l'exploration du champ des solutions consiste à se déplacer dans celui-ci afin de chercher la solution optimale sans avoir à évaluer toutes les solutions. Certaines méthodes, comme celle du *branch and bound*, permettent de trouver l'optimum global dans un champ de solutions sans avoir à en faire une exploration exhaustive. Ces méthodes ont besoin de propriétés particulières du champ pour permettre d'éliminer un sous-ensemble de celui-ci sans avoir à l'explorer. Il faut pour cela pouvoir assurer la croissance de la fonction objectif d'un sous-arbre de solution lors de l'exploration de celui-ci. La fonction objectif de notre problème n'ayant pas une telle propriété (elle nécessite l'élaboration complète d'une solution pour l'évaluer), les méthodes basées sur une connaissance partielle de la fonction objectif et de sa variation ne sont pas applicables.

L'algorithme proposé ici reprend le principe des algorithmes génétiques décrits dans [Forrest S., 1996]. En analysant la dynamique proposée par l'auteur, il est possible de distinguer six étapes dans un algorithme génétique. La dynamique complète est présentée dans la figure 51. Les six étapes listées dans l'article sont les suivantes :

- **Création d'une population initiale.**
- **Mécanisme de reproduction/sélection.**
- **Mécanisme de croisement.**
- **Mécanisme de mutation.**
- **Mécanisme de réparation (si besoin).**
- **Test d'arrêt.**

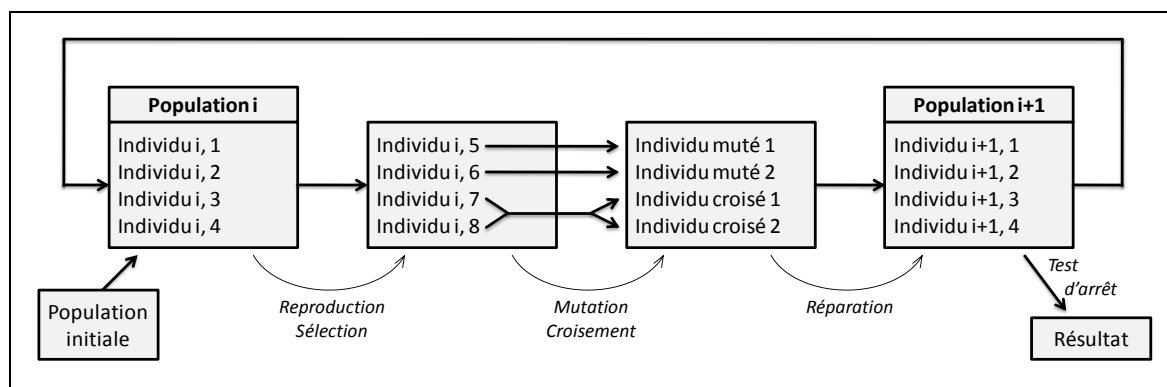


Figure 51 : Dynamique d'un algorithme génétique selon [Forrest S., 1996]

Dans l'algorithme proposé dans ce chapitre, un individu correspond à une topologie. Il est caractérisé par l'ordre des blocs dans l'installation. Cette partie du chapitre 4 va s'attacher à définir les six étapes proposées par [Forrest S., 1996] dans le cadre de l'optimisation de la topologie des ateliers de traitement de surface. Nous reprenons cet agencement des étapes et chercherons de définir chacune d'elles.

Note 1 : La ressource critique du problème présenté est le pont. Celui-ci assure tous les déplacements, et permet la fluidité du trafic au sein de l'atelier. L'optimisation de la topologie doit viser à réduire les déplacements de celui-ci. Cette solution est alors fortement dépendante de la production qui sera prise en compte. L'objectif final de la méthode est de permettre au pont d'être au bon endroit au bon moment, et ainsi limiter les longs déplacements à vide. C'est ce constat que l'on va exploiter dans les différentes étapes.

Note 2 : Au cours de l'algorithme, les populations seront de cardinal p .

3.1. Création d'une population initiale

Chaque individu de la population initiale sera généré de façon aléatoire (dans le champ des solutions acceptables) selon la méthode suivante.

A l'initialisation de la génération d'un individu, n blocs indépendants sont présents. Une connexion de type haut et une connexion de type bas sont sélectionnées. Dans le cas où la jointure des deux blocs par cette connexion est possible, les deux blocs sont joints. Dans le cas contraire, toutes les liaisons sont séparées et l'algorithme recommence avec les blocs initiaux.

A ce stade, il reste $(n - 1)$ blocs indépendants. Chacun de ces blocs est vérifié pour ne pas introduire de conflit avec les contraintes du problème (par exemple : vérification de la position en bout de ligne des entrées si cela est imposé). Si aucun conflit n'est observé, la jointure (paragraphe précédent) recommence avec ces $(n - 1)$ blocs. En revanche, si on constate la présence d'un conflit, les blocs sont séparés et l'algorithme recommence.

L'algorithme s'arrête lorsqu'il n'y a plus qu'un bloc. Cet algorithme est représenté dans la figure 52.

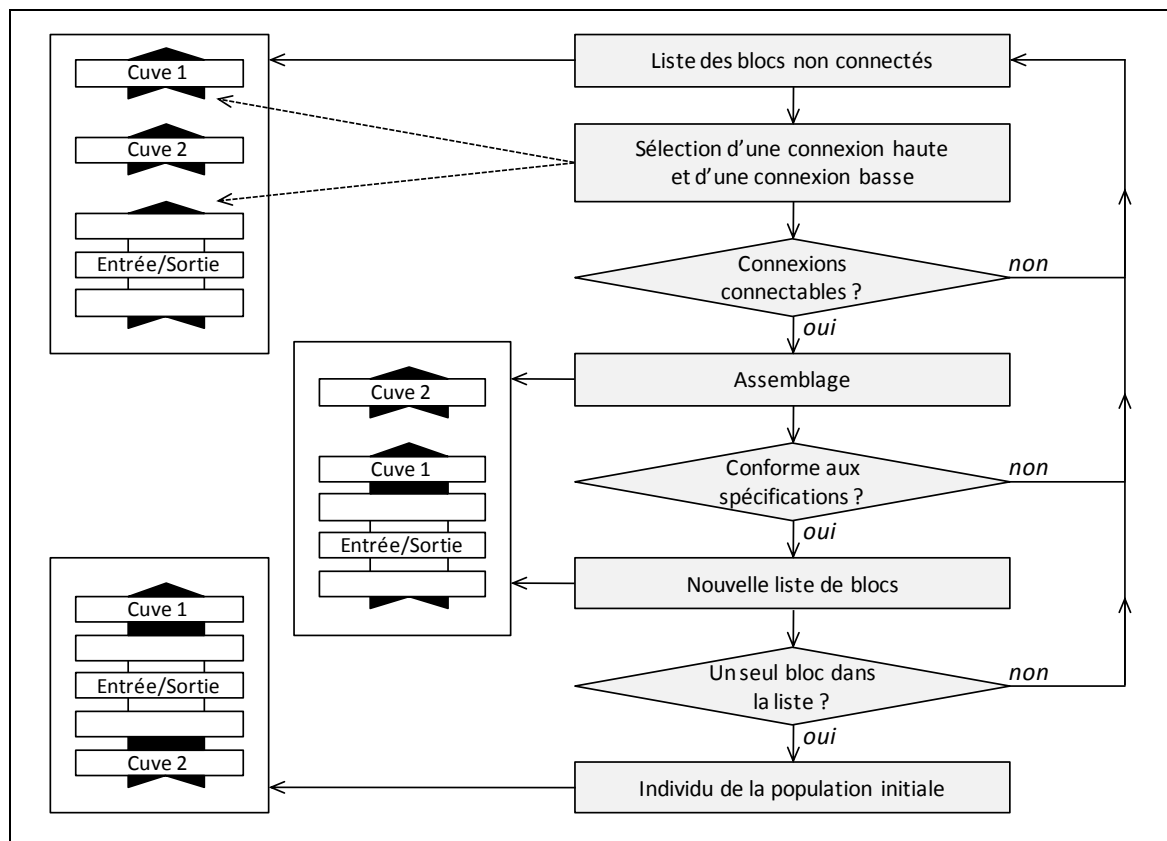


Figure 52 : Génération d'un individu de la population de départ

La population initiale est constituée de p individus sans lien entre eux. Ils ont peu de chances d'avoir des ressemblances (séquence commune de bains). Néanmoins, cette population initiale est constituée uniquement de topologies répondant au problème du placement des cuves.

3.2. Mécanisme de reproduction/sélection

La première étape du mécanisme de reproduction/sélection consiste à évaluer chacun des individus de la population (à l'aide d'une fonction appelée *fonction fitness*). Dans le cas du traitement de surface, une topologie est caractérisée, d'une part par sa capacité à réaliser complètement la

production, et d'autre part par ses performances. Pour ce faire, ces topologies sont soumises par simulation à une commande correspondant à une année de charge environ.

- Un blocage dans la simulation induit l'élimination de l'individu incapable de produire.
- Les individus capables sont comparés en fonction du temps total nécessaire pour réaliser la production donnée.

Cette étape aura pour effet d'éliminer la moitié des individus : soit en raison de leur incapacité à produire, soit pour leur faible performance relative. Dans le cas où beaucoup d'individus ont été éliminés pour leur incapacité à produire, la population est complétée par des individus issus de l'algorithme de génération de la population initiale. La population est ensuite dupliquée.

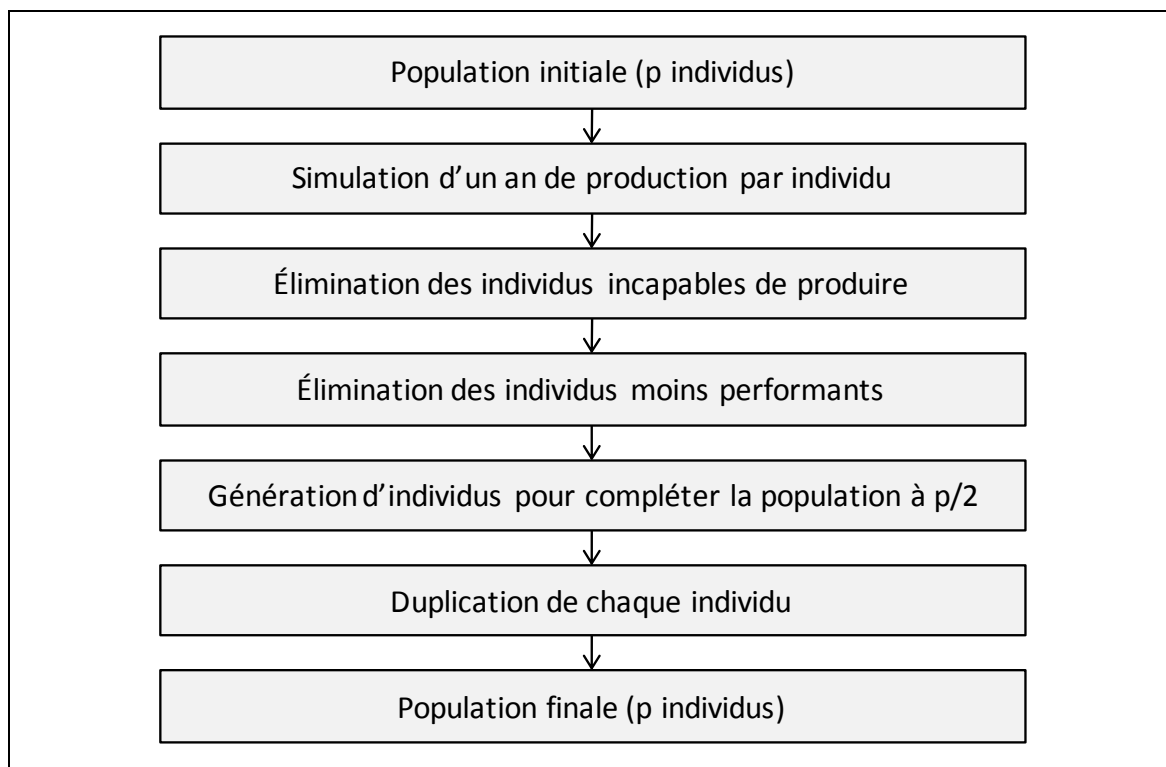


Figure 53 : Mécanisme de reproduction/sélection

Ce mécanisme conserve le nombre de membre de la population, en éliminant les moins performants et en introduisant parfois de nouveaux individus.

3.3. Mécanisme de croisement

L'objectif de croisement est d'obtenir un brassage des différents individus entre eux. Pour cela, il est nécessaire de choisir deux individus à croiser. Ce choix est fait aléatoirement parmi tous les individus de la population (pas de fonction d'attraction). Le problème de topologie doit, au fur et à mesure des pas de l'algorithme, faire ressortir des assemblages '*intéressants*' de postes de traitement. Pour y parvenir, nous proposons d'utiliser le *mécanisme de croisement* de l'algorithme, qui fera ressortir des enchainements de postes performants. En effet, deux postes côte à côte peuvent signifier qu'il y a de nombreux échanges entre eux. Si deux individus (topologies parentes) disposent de cette même

séquence, il est possible qu'elle soit judicieuse. Les deux individus parents sont donc comparés, pour reconnaître les séquences semblables de postes. De telles séquences sont alors dites *intéressantes* puisque les deux parents en disposent. L'algorithme de reconstitution imposera alors aux individus issus de ce croisement de disposer également de ces séquences. Les itérations successives de l'algorithme permettront de faire naître des séquences intéressantes de plus en plus nombreuses et de plus en plus longues. La génération de deux individus à travers ce mécanisme est représentée dans la figure 54.

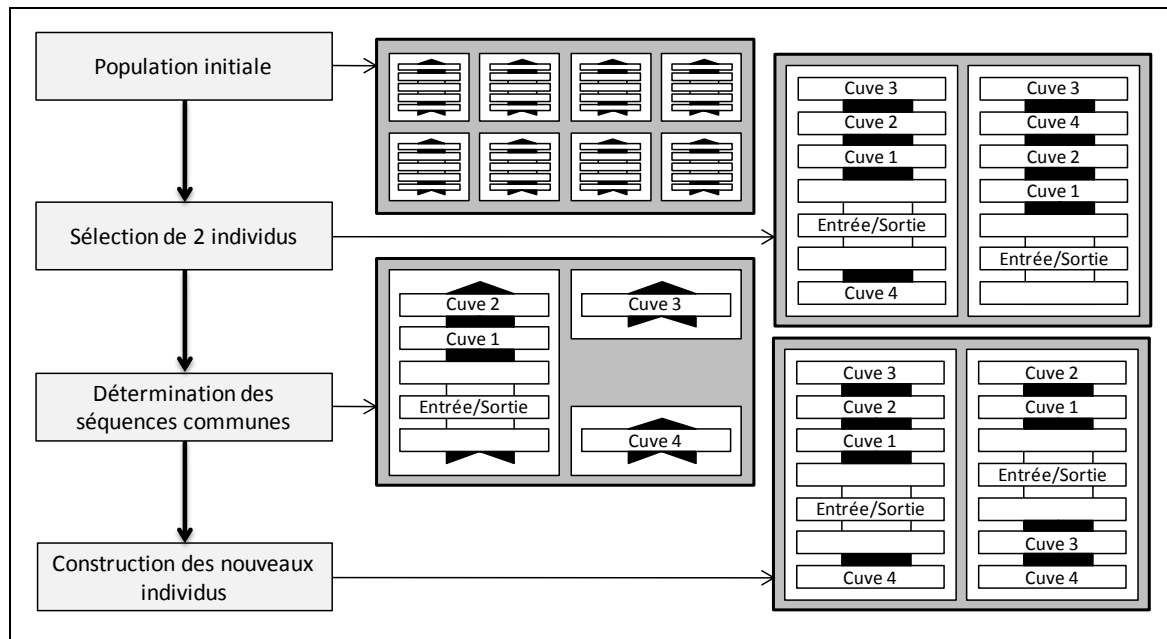


Figure 54 : Mécanisme de croisement

Le mécanisme de croisement est utilisé pour reconstituer un lot d'individus de taille égale à 95% de la population initiale.

3.4. Mécanisme de mutation

L'objectif d'un mécanisme de mutation est de réintroduire de nouveaux enchaînements dans la population sans renouveler tous les enchaînements acquis par sélection naturelle. Il a pour principe de prendre un individu, de le couper aléatoirement en m blocs et de réassembler ceux-ci. La sélection des deux points de séparation se fait aléatoirement sans préférence. La reconstruction se fait par l'algorithme de génération d'un individu de la population initiale. Cet algorithme est appliqué en utilisant les m blocs séparés comme donnée source.

Le mécanisme de mutation est utilisé pour générer 5% de la population initiale. Ainsi, la conjonction du mécanisme de croisement et du mécanisme de mutation permet de reconstituer une population brassée de même cardinal que la population initiale.

3.5. Mécanisme de réparation

Le mécanisme de réparation d'un algorithme génétique est utilisé pour réparer les solutions issues du mécanisme de croisement et de mutation. Cette réparation a lieu dans le cas où la solution construite ne serait pas possible. Ce mécanisme rend possible une solution impossible.

Dans le cas des travaux présentés ici, les algorithmes de croisement et de mutation génèrent des topologies possibles. Certaines des topologies mécaniquement faisables peuvent néanmoins montrer que leur mise en production induira des blocages. Ces solutions sont supprimées lors du mécanisme de reproduction/sélection.

3.6. Test d'arrêt

Chaque population est constituée de p individus. Ces individus ont été évalués lors de la phase de sélection/reproduction. A l'issue de cette phase, la meilleure des implantations est comparée avec la meilleure implantation de la population précédente. Si le meilleur individu de la population $i + 1$ montre une meilleure performance que le meilleur individu de la population i , l'algorithme continue en s'initialisant avec la population $i + 1$. Dans le cas contraire, l'algorithme s'arrête et retourne comme résultat le meilleur individu de la population i .

4. Application à un cas industriel

Cet algorithme a été appliqué pour la construction de la topologie du *problème industriel B* présenté dans le chapitre 3 (voir figure 31). Le problème détaillé sera présenté dans cette partie avec ses contraintes propres, puis les performances de l'algorithme sur cet exemple seront analysées. Cette partie s'achèvera par quelques résultats expérimentaux comparés à une implantation d'expert.

4.1. Présentation du cas industriel

L'installation industrielle B, représentée dans la figure 31, sur laquelle cette étude a été validée, est constituée de 25 blocs élémentaires connectables entre eux :

- 3 postes d'entrée/sortie
- 2 navettes
- 4 postes d'attente
- 16 cuves (dont une cuve d'échange)

Afin de répondre au cahier des charges proposé par l'industriel (notamment dû aux contraintes de dimensions et d'interfaçage avec l'extérieur de l'installation), les cuves et les postes d'attente se retrouveront sur une seule ligne desservie par deux ponts. Cette ligne est séparée en deux par une zone commune aux deux ponts. Cette zone est constituée d'une navette et d'un poste d'échange. La ligne de traitement commencera par une navette et se terminera par un poste de déchargement. Ces

contraintes permettent un regroupement de certains blocs. Ces **blocs agrégés** sont utilisés pour l'initialisation de l'algorithme d'implantation (voir figure 55). Ces regroupements réduisent à 20 ! le nombre de topologies différentes (soit plus de 10^{18} topologies).

Les cuves de traitement sont presque toutes liées à une cuve de rinçage dans les gammes de traitement. Cette remarque pourrait être prise en compte pour agréger les blocs deux par deux. Cette décision n'a pas été prise afin de laisser le maximum de latitude à l'algorithme d'implantation.

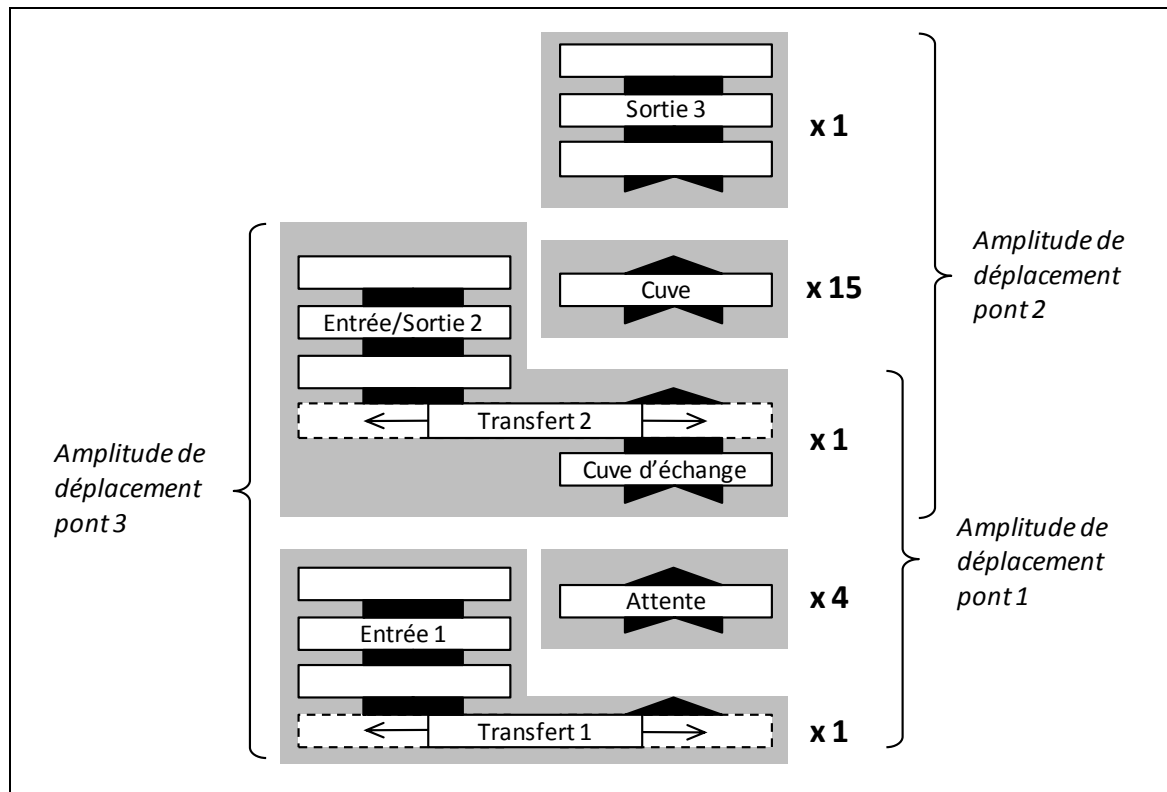


Figure 55 : Blocs élémentaires à planter pour le cas industriel B

Le **flux d'arrivée** des produits dépend de la localisation du premier poste de sa gamme de traitement. Ainsi un produit dont la première opération est desservie par le pont 1 entrera dans l'installation via l'entrée 1. De même, un produit dont la première opération est desservie par le pont 2, entrera via l'entrée 2. A la **fin d'un traitement**, un produit dont la dernière opération est desservie par le pont 1, sortira par la sortie 2, tandis qu'un produit dont la dernière opération est desservie par le pont 2, sortira par la sortie 3. Enfin, un produit traité par le pont 1, dont la prochaine opération se trouve desservie par le pont 1, sera déposé dans la cuve d'échange par le pont 1 puis prise par le pont 2. Ces contraintes permettent de favoriser la fluidité du système.

4.2. Exécution de l'algorithme

L'algorithme génétique présenté a été implanté dans *Microsoft Visual Studio 2005*. Ce programme prend comme entrée *une production annuelle type* et *les blocs présentés avant*, dans l'objectif de générer une topologie. L'évaluation des performances de chaque topologie se fait par simulation sous *Siman/Arena*. Le grand nombre de topologies différentes testées impose une génération

automatique des modèles de simulation. Cette génération a été directement implantée dans le programme *Visual Studio 2005* en utilisant la communication *COM* avec *Siman/Arena*. L'implémentation des différents algorithmes présentés précédemment sont représentés dans la figure 56.

Afin d'obtenir des résultats représentatifs de la qualité de chaque topologie testée, celles-ci sont soumises à 20 scénarii de production. Ces productions reprennent les quantités produites durant une année mais diffèrent sur la répartition temporelle de celle-ci (regroupement par lot, lancement cyclique, lancement aléatoire). La moyenne du temps de production des 20 réplifications est utilisée comme indicateur de performance de la topologie.

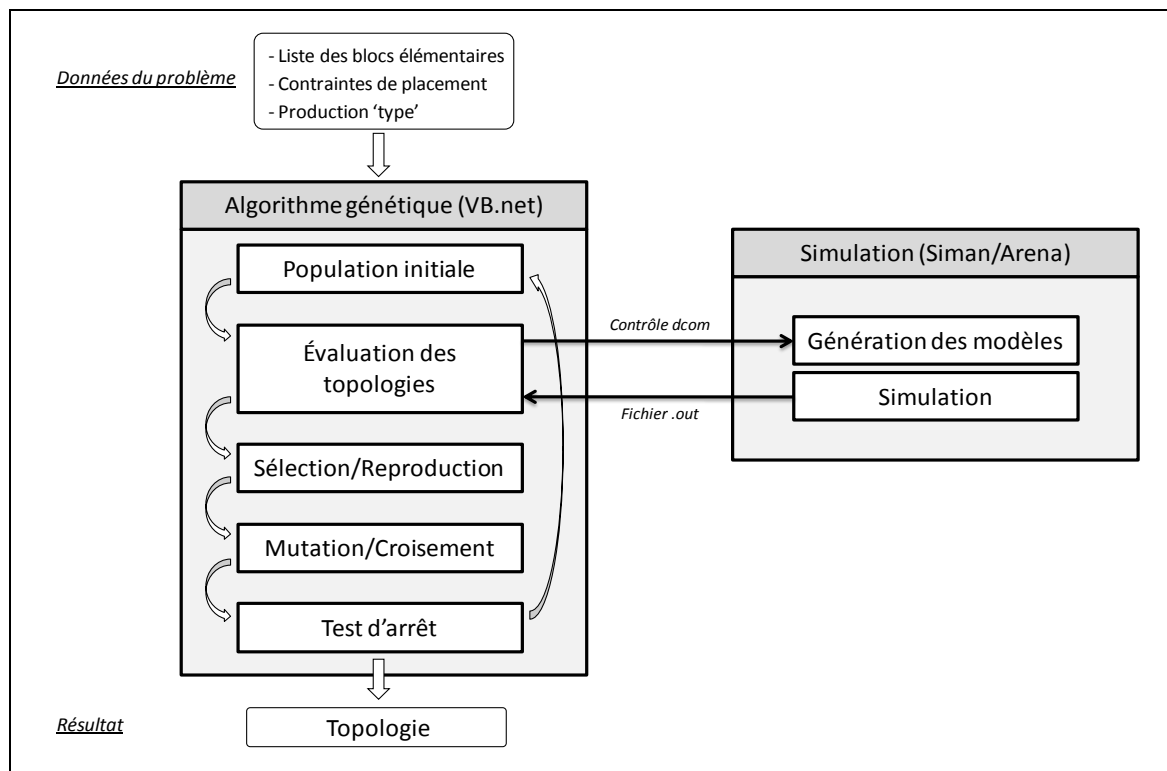


Figure 56 : Implémentation de l'algorithme génétique

4.3. Résultats

L'objectif de l'algorithme génétique est de proposer la meilleure topologie vis-à-vis de la production donnée. Il est donc intéressant d'analyser l'évolution du meilleur individu (meilleure topologie) de chaque population. Une des séries des résultats obtenus est reprise dans la figure 57.

La première remarque à faire est la **convergence rapide de l'algorithme** (ou du moins l'occurrence du test d'arrêt). La meilleure solution trouvée par cette instance est située dans la cinquième génération, ce qui est rapide pour un algorithme génétique, mais normal compte tenu du faible nombre d'inversions possibles. La suppression du test d'arrêt pour analyser quatre générations supplémentaires (Géné. 7 à Géné. 10) montrent une oscillation des performances de ces générations entre les résultats de la quatrième et de la cinquième génération. Cette analyse montre une certaine efficacité du test d'arrêt proposé.

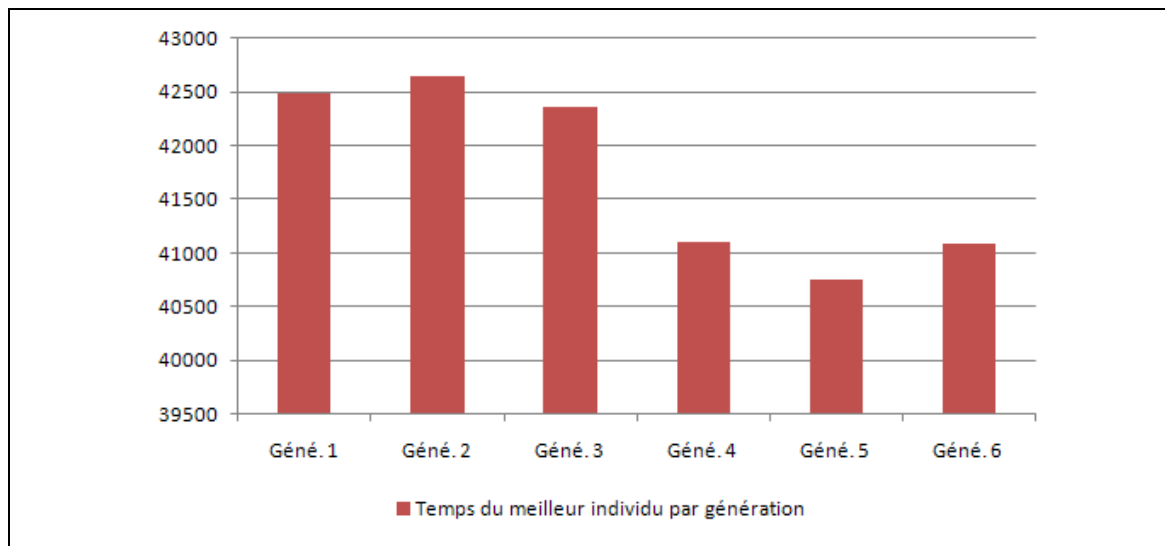


Figure 57 : Évolution du meilleur individu de chaque génération (unité arbitraire)

L'amélioration des meilleurs individus entre la génération 2 et la génération 5 montrent l'efficacité de la **fonction de croisement**. En analysant les topologies, on remarque la création, petit à petit, de séquences regroupant un traitement avec son rinçage associé. La diminution des performances entre la génération 1 et la génération 2 est due à la non existence, dans la première génération, de séquences communes à plusieurs individus. Le caractère aléatoire des individus de la première génération pourrait être corrigé par la création de séquences communes, mais les meilleures performances ne sont pas prouvées.

Le **gain de performances** de la meilleure topologie issue de l'algorithme par rapport à la meilleure topologie de la première génération est de 4,6%. Sur une année complète de travail (fonctionnement 24 heures sur 24), cela représente un gain de 16 jours en capacité de l'installation.

Ce problème a été proposé à **un expert** du traitement de surface qui réalise habituellement la topologie de l'installation industrielle. La topologie issue de l'algorithme montre un gain de 1,5% par rapport à la topologie proposée par l'expert sur le même problème. La différence entre les deux topologies réside dans le caractère logique ou intuitif de la topologie proposée par l'expert : les cuves sont positionnées les unes à côté des autres, conformément aux gammes de traitement les plus fréquentes. L'algorithme, par son absence de logique, permet de produire des topologies légèrement meilleures en terme de productivité, mais moins organisées : par exemple le rinçage d'un traitement est parfois avant, parfois après celui-ci. La topologie permet alors de réduire les déplacements du pont à vide.

Afin de mesurer les performances de cet algorithme par rapport à la **génération de topologies aléatoires**, nous avons construit une population aléatoire d'une taille équivalente à six générations. Les performances de toutes ces topologies ont été évaluées. La meilleure de ces topologies produit l'ensemble des pièces en 41630 unités de temps. Ce résultat est moins bon que celui de l'expert et que celui de l'algorithme. Cette conclusion n'est pas systématique car, en théorie, la génération aléatoire permettrait de trouver la topologie optimale. Cependant, au vue de la petite taille des solutions explorées par rapport à l'immensité du champ de topologies, il est très peu probable que la meilleure topologie soit trouvée sans orienter la recherche de celle-ci.

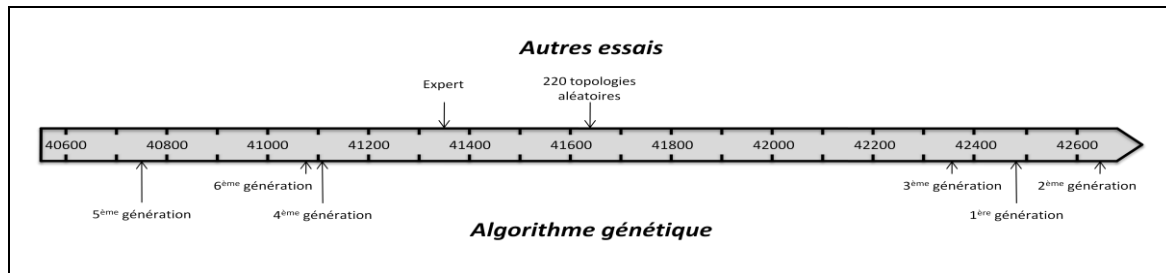


Figure 58 : Comparaison des différentes topologies

Les différentes solutions de génération de topologie, et leur position relative, sont représentées dans la figure 58.

5. Conclusion

Dans ce chapitre, nous avons exposé un premier problème connexe à celui de l'ordonnancement des lignes de traitement de surface : la génération automatique de la topologie du système. Nous y avons présenté une méthode systématique et automatisable de génération de topologie, basée sur un algorithme génétique. L'application de cette méthode à un problème particulier a permis, dans cet exemple, de montrer l'efficacité de l'algorithme et la pertinence des fonctions de croisement, mutation, (...) choisies. Au final, sur l'exemple présenté, cette méthode a permis une augmentation de 1,5% de la productivité. Dans le cadre d'une implantation initiale d'atelier, cette méthode ne demande que très peu d'investissement et permet de générer une topologie sans expert. Dans le cadre d'une réimplantation d'une installation existante, il faut comparer le gain en productivité (1,5% équivaut à 5,5 jours de production par an) à l'investissement nécessaire à la réimplantation (coût de non production et coût de réimplantation).

Chapitre 5 : Aide au lancement des produits dans un RHSP

| | |
|---|------------|
| 1. La prise de décision dans un système réactif | 100 |
| 1.1. Exemple avec 2 produits | 100 |
| 1.2. Exemple avec n produits | 101 |
| 2. Heuristique d'exploration | 106 |
| 3. Résultats | 108 |
| 3.1. Résultat de l'heuristique | 108 |
| 3.2. Étude du nombre de pas d'optimisation | 110 |
| 3.3. Étude du lancement successif de plusieurs instances de l'heuristique | 111 |
| 4. Conclusion | 113 |
| 5. Annexes : résultats du problème 2 | 114 |

Nous avons présenté dans les parties précédentes un pilotage réactif d'une installation de traitement de surface. Un des inconvénients majeurs d'un tel mode de pilotage est l'absence de prévision. Cette absence conduit à l'impossibilité d'optimiser le flux au sein de l'atelier. La figure 28 a mis en évidence trois leviers d'optimisation de la production : le premier levier intervient sur le dimensionnement et l'implantation de l'atelier pour permettre une bonne fluidité des produits ; le second modifie l'ordonnancement des tâches ; le troisième agit sur l'entrée des pièces dans l'atelier. Dans certains modes de pilotage, notamment le pilotage prédictif, l'optimisation de l'ordonnancement ne concerne que les deux derniers leviers. Dans un mode réactif, le pilotage ne modifie pas l'arrivée des pièces, mais les prend en compte au fur et à mesure de leurs arrivées. Il peut choisir entre quelques pièces prêtes à produire, mais le choix se fait rarement sur des critères d'optimisation globale.

Dans ce chapitre, nous proposerons une réflexion sur l'aide au lancement des pièces dans un pilotage réactif des installations de traitement de surface. L'objectif est de définir, pour une production donnée, un ordre de passage des produits dans l'installation qui permet une bonne productivité.

Un système de production muni d'un pilotage réactif peut s'apparenter à un système imprévisible, voire à un système chaotique au sens défini par Poincaré dans [Poincaré H., 1892]. Dans un tel système, une petite modification peut avoir de grandes conséquences sur l'ordonnancement. Ces conséquences auront un impact sur la qualité et la productivité.

Les auteurs de [Fleury G. *et al.*, 1999] ont travaillé sur la génération d'un planning permettant de prendre en compte de petites perturbations. Le planning généré donne une plus grande stabilité au

système de production muni de son pilotage. Cependant, l'ordonnancement reste statique, donc inadapté aux perturbations de grande amplitude.

Dans ce chapitre, nous nous intéresserons à l'impact de l'ordre de lancement des produits dans le système de production sur la productivité de celui-ci. Cet ordre permet de choisir les produits dans un lot de produits à traiter, ou d'informer l'atelier précédent de l'ordre des produits à préparer.

1. La prise de décision dans un système réactif

Dans [Cardin O. *et al.*, 2005] et [Cardin O. *et al.*, 2008], les auteurs ont cherché à mesurer l'impact de la prise d'une décision dans un système réactif. L'application présentée traite du pilotage des systèmes de production flexibles. Les auteurs mettent en évidence, lors d'une décision de routage, l'impact que peut avoir une mauvaise décision sur la productivité du système. Dans ce contexte, un pilotage purement réactif ne permet pas de comparer plusieurs décisions et leur impact sur les performances finales du système manufacturier. Pour résoudre ce problème, les auteurs proposent d'utiliser la simulation en ligne afin d'évaluer les performances produites par chacune des décisions, dans le but d'appliquer celle qui aura les meilleurs conséquences. Une seconde application de ces travaux est présentée dans [Cardin O., 2007]. Cette application aborde la prise de décision avant le lancement de production. Dans ses travaux, l'auteur explique qu'une bonne préparation de la production (choix de la date d'introduction, regroupement des produits...) peut modifier les performances globales.

1.1. Exemple avec 2 produits

Dans ce chapitre, nous abordons le cas de l'ordre du lancement des pièces dans un système de production. Prenons par exemple une installation de traitement de surface à 2 postes d'entrée et deux cuves (une cuve de traitement et une cuve de rinçage). Dans l'exemple représenté dans la figure 59, deux produits doivent passer dans la cuve de traitement et la cuve de rinçage avec des temps de traitement différents.

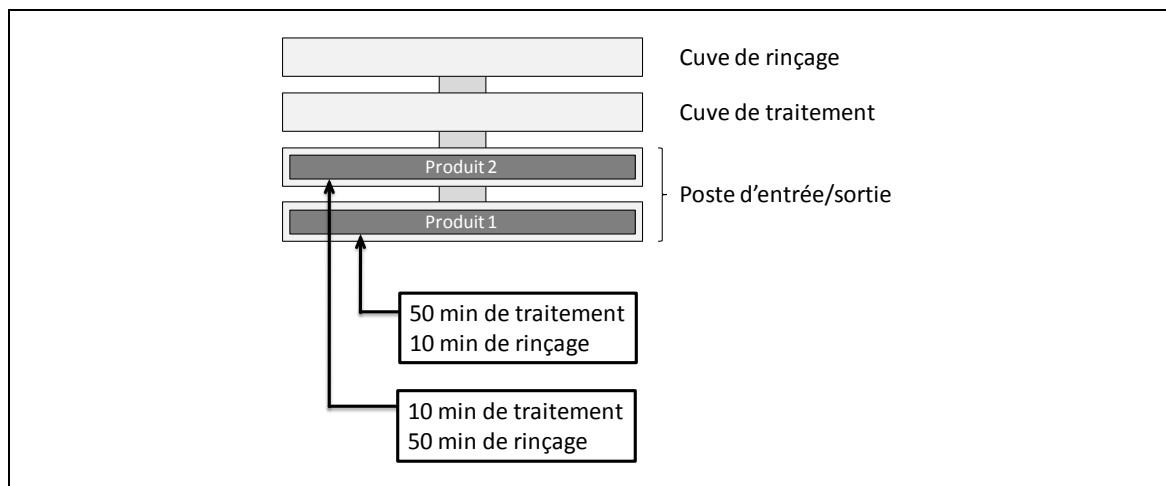


Figure 59 : Exemple de lancement de 2 produits

Dans cet exemple élémentaire, le pont a deux possibilités : déplacer le produit 1 en premier ou déplacer le produit 2 en premier. Compte tenu des gammes de traitement différentes, les deux ordonnancements n'auront pas le même *makespan*. Les deux ordonnancements émergents d'un pilotage dynamique sans optimisation sont représentés dans la figure 60.

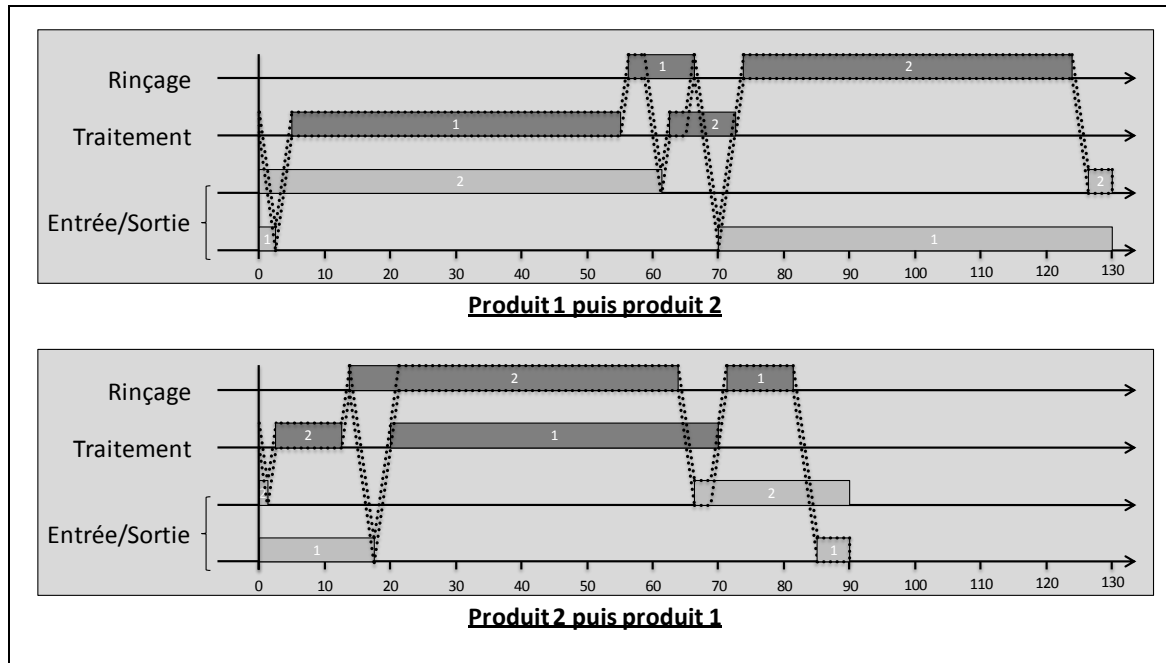


Figure 60 : Exemple d'ordonnement de 2 produits

Cet exemple montre que l'ordre de lancement des produits peut avoir une influence dans les performances du système. Dans un cas la production se termine au bout 127 minutes, dans l'autre cas la production se termine en 85 minutes.

1.2. Exemple avec n produits

La comparaison des deux ordonnancements présentés dans le paragraphe précédent doit être prise avec précaution : l'ordonnement n'est constitué que d'une phase de chargement et d'une phase de déchargement. Dans cette partie, nous proposons d'étudier tous les ordres possibles de lancement d'une production constituée d'un plus grand nombre de produits.

1.2.1. Définition des exemples

Nous étudions deux productions typiques lancées dans l'installation industrielle A présentée précédemment. La première production est constituée de quatre gammes, tandis que la seconde production est constituée de 3 gammes. Ces deux exemples couvrent la production d'environ 8 heures.

| | Exemple 1 | Exemple 2 |
|-----------|-----------|-----------|
| Produit A | 3 | - |
| Produit B | 3 | - |
| Produit C | 3 | - |
| Produit D | 3 | 5 |
| Produit E | - | 2 |
| Produit F | - | 2 |

Tableau 10 : Répartition de la production dans les deux exemples étudiés

La première étape, pour analyser l'impact d'un changement d'ordre des produits à l'entrée de l'installation, est de construire un ordonnancement pour l'ensemble des ordres de lancement possibles. Pour l'exemple 1, il existe 369.600 ordres de lancement différents [$12! / (3! \times 3! \times 3! \times 3!)$] tandis que pour l'exemple 2, il existe 756 ordres de lancement différents [$9! / (5! \times 2! \times 2!)$].

1.2.2. Montage expérimental

Un des points qui nous semble intéressant dans cette étude est d'obtenir la répartition du temps de production en fonction de l'ordre de lancement. Cette répartition est obtenue par évaluation exhaustive de tous les ordres possibles de lancement des produits. Pour évaluer le temps probable de traitement d'un lot de produit, nous utilisons le modèle de simulation à événement discret implémentant les règles de pilotage définies dans le chapitre 3.

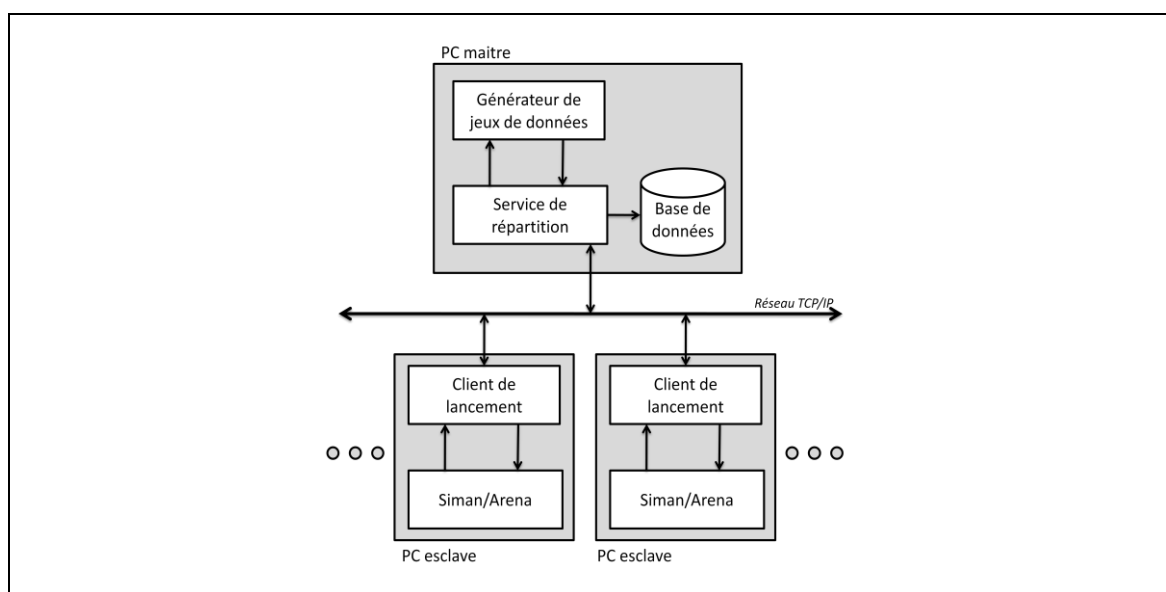


Figure 61 : Architecture réseau de la répartition des simulations

Sachant qu'il faut quelques secondes pour simuler un ordre de lancement (temps utilisé à 99% par le chargement de *Siman*), la simulation exhaustive de tous les ordres des produits aurait représenté plus d'un mois de simulation. Afin de réduire au minimum le temps de simulation, et en partant du constat que les simulations sont indépendantes les unes des autres, nous avons développé une architecture de répartition des simulations sur 30 machines reliées par un réseau informatique (voir figure 61).

Afin de développer une architecture générique et réutilisable pour d'autres applications, nous avons développé un protocole réseau au niveau de la couche applicative du modèle OSI publié dans [IEC 7498-1, 1994]. Le protocole utilisé n'utilise pas de couche de session ni de couche de présentation, ce qui le rend vulnérable en environnement informatique ouvert.

Afin de ne pas avoir à configurer les clients en cas de changement de maître, et pour permettre à plusieurs utilisateurs d'utiliser les clients, un second **protocole d'auto-configuration**, basé sur les échanges multicast, permet aux clients de découvrir tous les maîtres du réseau.

La **diffusion multicast** a pour principe d'envoyer une information à des destinataires sans les connaître à l'avance. Le publieur envoie des paquets *UDP/IP* vers une adresse IP virtuelle particulière (adresse de diffusion). Les clients intéressés par un flux multicast informent le matériel de routage de leur intérêt pour le flux (abonnement). De son côté, le matériel de routage duplique le flux pour le diffuser vers tous les abonnés. Dans des tels échanges, l'adresse réseau est attribuée au type d'information indépendamment du/du l'annonceur(s) et du/du abonné(s). Dans un tel protocole, il y a une seule adresse configurée (l'adresse de diffusion) ; zéro, un ou plusieurs annonceurs ; zéro, un ou plusieurs abonnés.

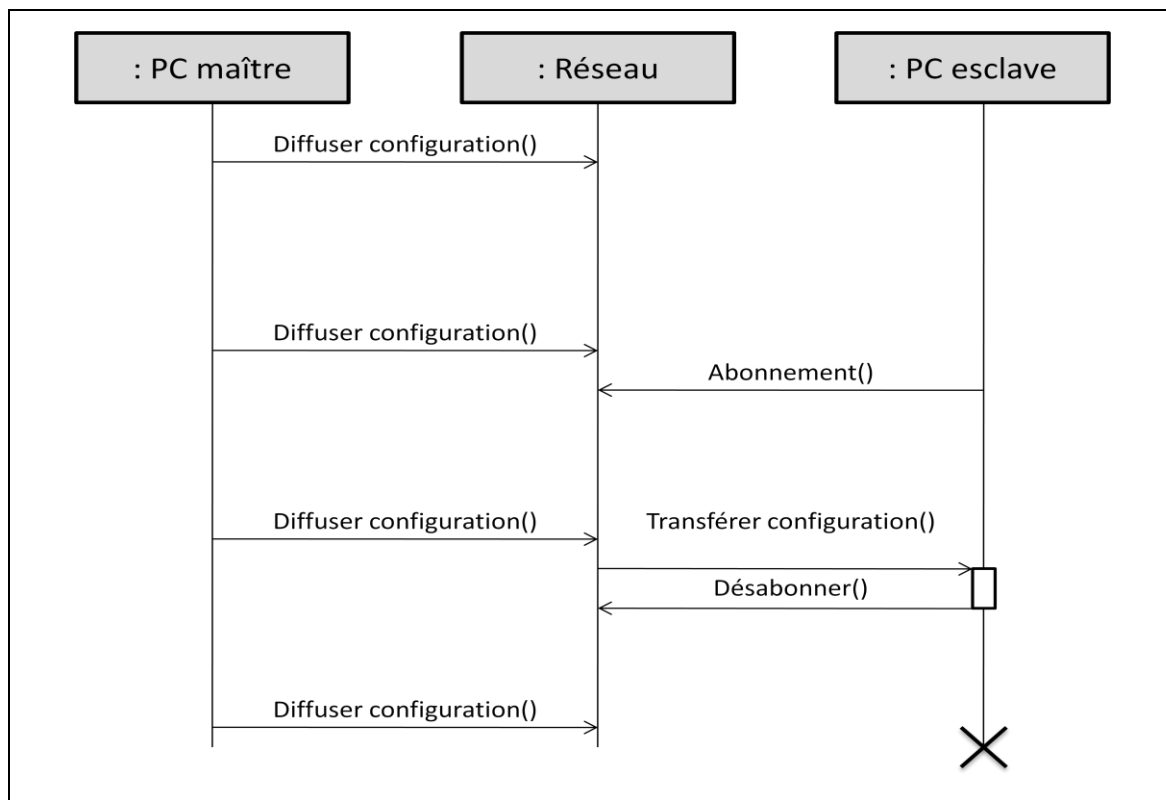


Figure 62 : Protocole d'auto-configuration des esclaves basé sur un flux UDP/IP multicast

Dans notre cas, le (ou les) PC maître diffuse(nt) sur une adresse multicast sa configuration (IP, port, service recherché). Les clients, de leur côté, s'abonnent à ce flux pour rechercher les PC maîtres à servir. La séquence d'auto-configuration est décrite dans la figure 62. Ce protocole a l'avantage de mettre le minimum de configuration sur le client.

Une fois l'étape d'auto-configuration franchie, les clients contactent le serveur à l'aide des paramètres de configuration reçus. Le serveur de répartition leur attribue une tâche. Une tâche est constituée d'un ou plusieurs fichiers (modèle de simulation, jeu de données), et d'une commande à exécuter. Une fois l'exécution terminée, le serveur va récupérer sur le client les résultats de la tâche. Nous utilisons ici un protocole proche du protocole *FTP* pour le transfert des fichiers et du *Telnet* pour l'exécution à distance d'un programme. Le diagramme de séquence de ces échanges est repris dans la figure 63.

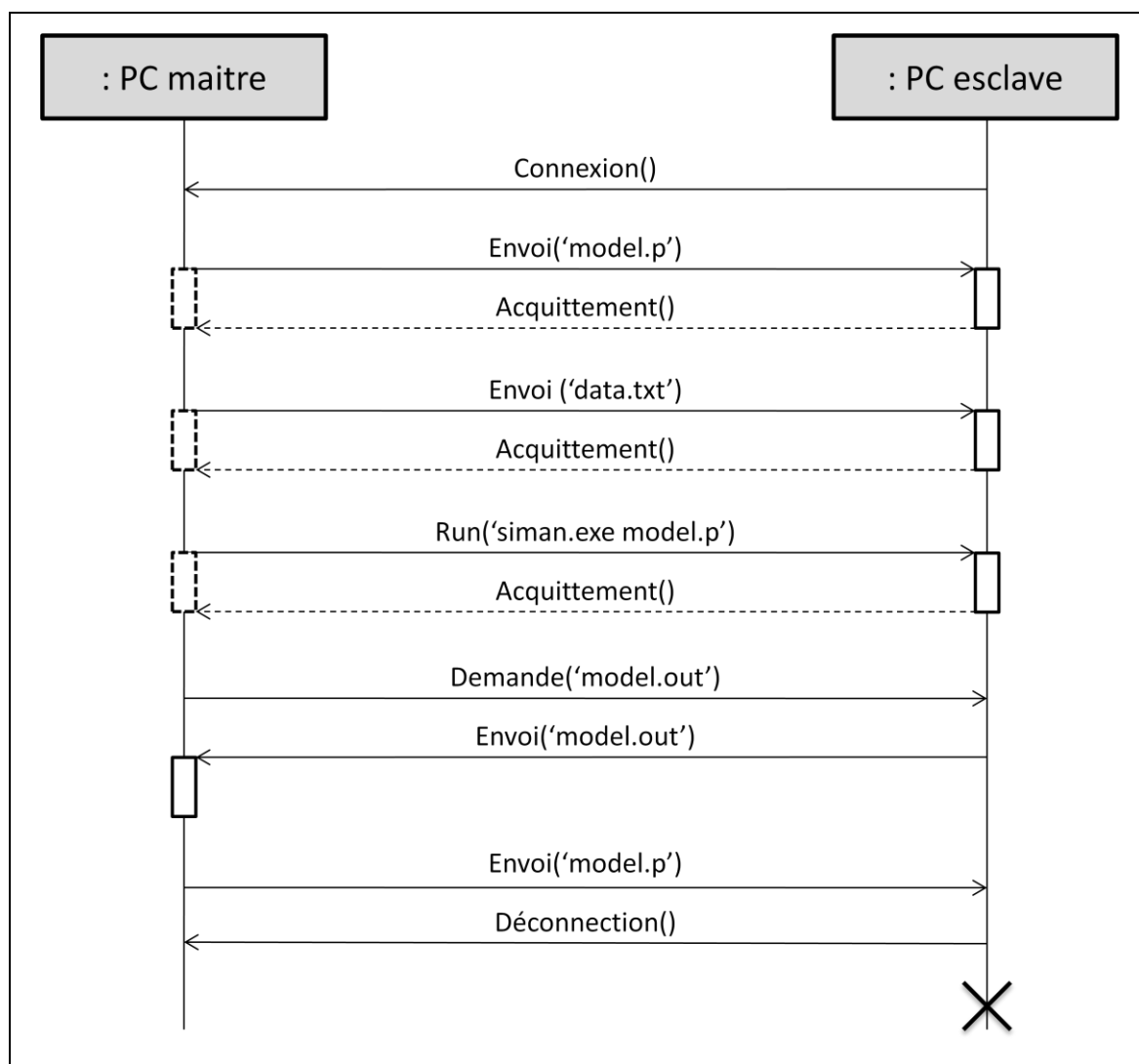


Figure 63 : Protocole d'exécution d'une simulation sur un esclave

Le montage réseau expérimental décrit dans ce paragraphe nous a permis d'évaluer tous les ordres de lancement des produits des deux problèmes évoqués. Le protocole d'auto-configuration permet d'ajouter ou d'enlever des esclaves en fonction de la disponibilité de ceux-ci. Le protocole d'exécution à distance permet de répartir les demandes sur les différents esclaves.

1.2.3. Résultats

La structure de *simulation répartie*, présentée précédemment, nous a permis de réduire la durée initiale nécessaire à l'évaluation de tous les ordres de lancement. Cette durée est passée (pour les deux problèmes) de plus d'un mois à moins de 2 jours. De plus, le caractère générique de la structure développée la rend réutilisable pour d'autres applications. Seul le serveur doit être modifié en fonction des besoins.

L'objectif qui nous intéresse dans le problème du traitement de surface est d'avoir la meilleure productivité. Dans le cas d'une production donnée (nombre de produits et répartition des gammes), la productivité et le temps de production sont liés. Nous proposons de représenter dans nos graphiques la répartition des différentes instances de simulation en les regroupant par temps de production. L'augmentation de la productivité revient à chercher un temps de production faible.

Note : Afin de ne pas surcharger ce rapport, seuls les résultats du problème 1 seront intégrés ici. Les résultats du problème 2 sont disponibles en annexe à la fin de ce chapitre (page 114). Un tableau de correspondance entre les résultats du problème 1 et les résultats du problème 2 est disponible (tableau 11 page 114).

La répartition des instances par temps de production est représentée dans la figure 64 pour le problème 1 (et rapportés en figure 74 page 114 pour le problème 2)

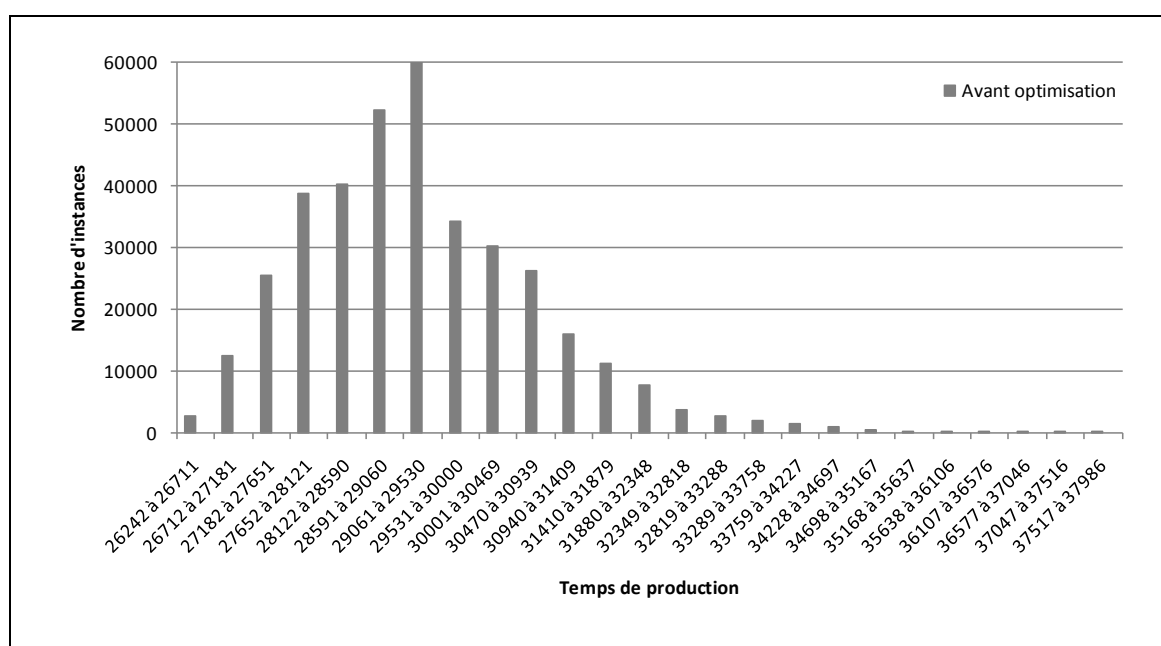


Figure 64 : Répartition des 369 600 instances du problème 1 par temps de production

Dans le problème 1, on remarque que l'ordre le plus défavorable (situé dans la droite du graphique) aura pour effet de faire chuter la productivité de 45% par rapport à l'ordre de lancement le plus favorable (à gauche dans le graphique) ! Cette perte de productivité entre le meilleur et le pire des ordres est de 22% pour le problème 2. Ce constat confirme l'importance du choix de l'ordre de lancement des produits dans l'installation.

Un tirage aléatoire de l'ordre de lancement des produits pour le problème 1 aura une productivité située en moyenne à 12% de l'optimum.

2. Heuristique d'exploration

Le constat qui nous a permis de proposer l'heuristique présentée dans cette partie est issu de la figure 60 (page 101) : une inversion de deux produits dans l'ordre de lancement permet, dans certains cas, un gain non négligeable en productivité. L'heuristique présentée ci-après est une heuristique de descente (voir figure 65) par exploration du voisinage dans le champ des ordres de lancement possible.

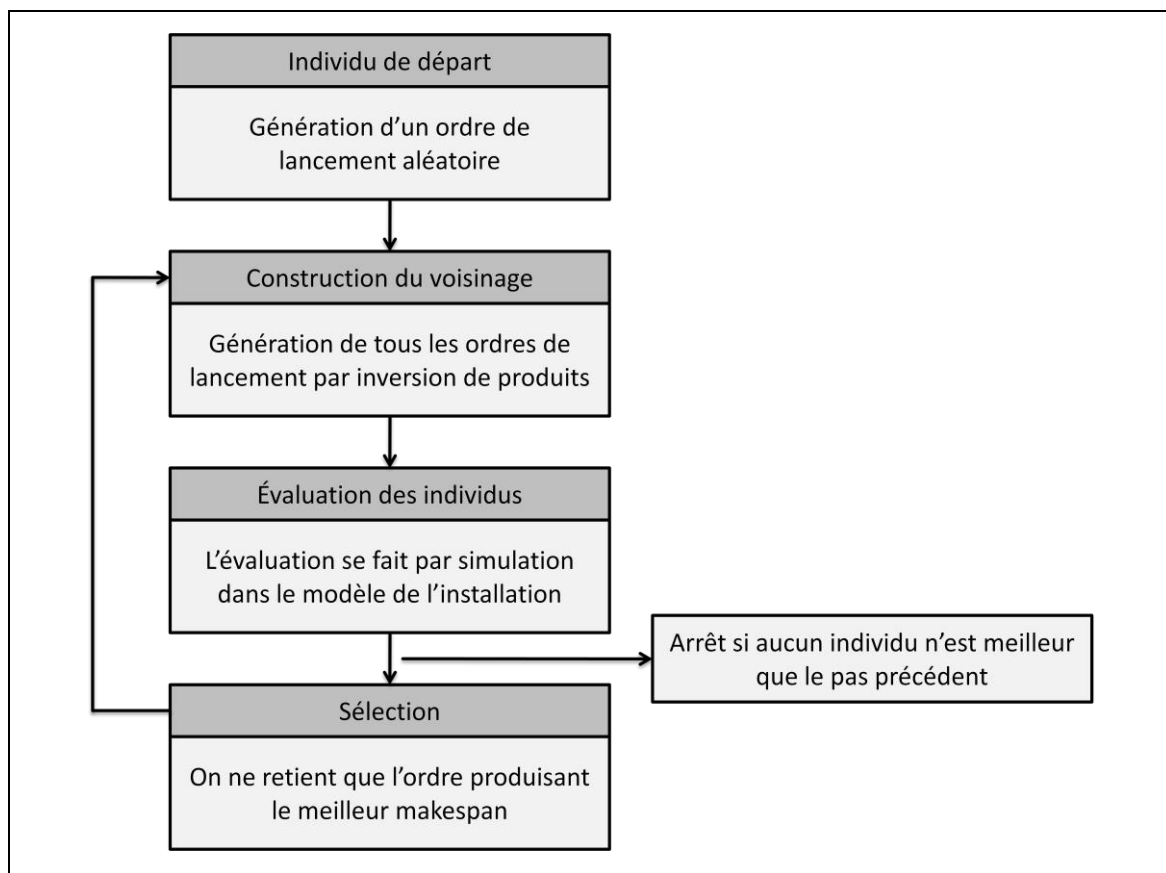


Figure 65 : Heuristique de descente

Le voisinage d'une solution est l'ensemble des ordres de lancement, chacun généré par inversion de deux produits. Un ordre de lancement constitué de n produits à lancer aura, au plus, $[n(n+1)/2]$ voisins (il y aura moins de voisins dans le cas de plusieurs produits de même gamme qui produiront les mêmes performances). On peut tout de suite remarquer que l'exploration exhaustive des voisins, dans le cas d'un problème avec beaucoup de produits à lancer, devient vite insurmontable (bien que de complexité polynomiale). Nous proposerons un peu plus loin de n'explorer qu'une partie des voisins de chaque solution (pour réduire encore cette complexité).

L'heuristique simple, telle que présentée ici, peut donner un optimum local loin de l'optimum global. Nous verrons dans les exemples appliqués que l'optimum local trouvé est très souvent proche de

l'optimum global. D'autre part, il est possible d'augmenter cette heuristique par une recherche Tabou. Compte tenu des performances de cette heuristique simple, ce développement n'est ici pas nécessaire.

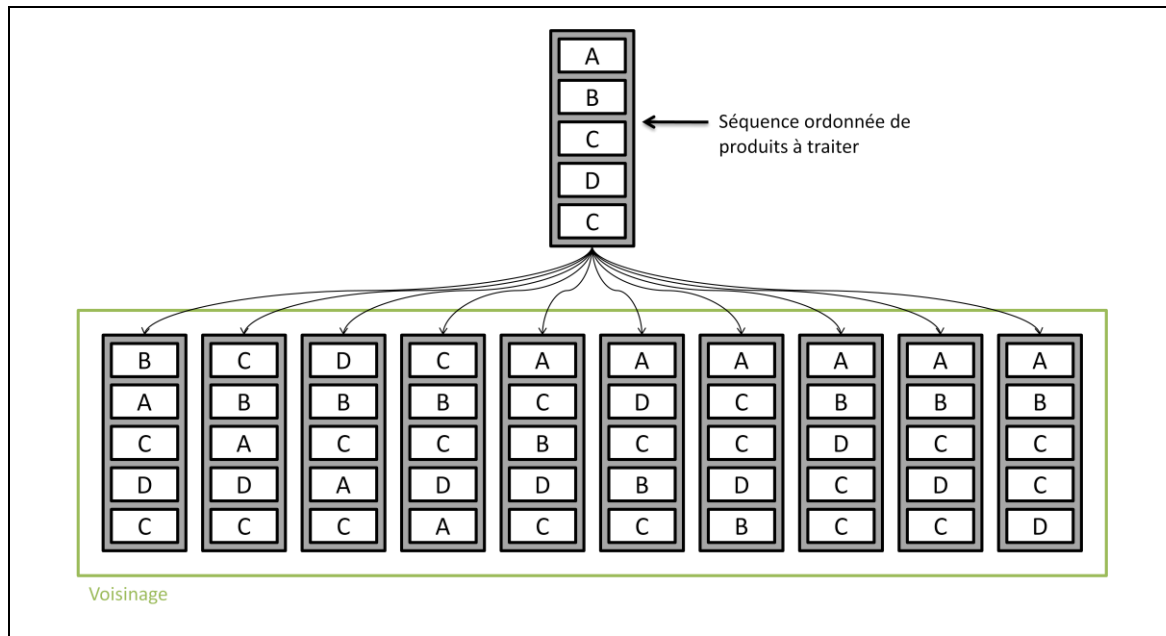


Figure 66 : Exemple de voisinage d'une séquence de lancement

Il n'est pas nécessaire d'augmenter le champ des voisins par des inversions plus complexes car, dans une certaine mesure, une double inversion revient en général à deux inversions successives. La simplification des inversions induit une augmentation du nombre de pas de descente. Seul le cas où une double inversion fait intervenir le même produit, (inversion circulaire de trois produits) n'est pas pris en compte.

Afin de palier aux limites évoquées ici, nous proposons 4 scénarii pour cette heuristique, permettant d'étudier l'influence des deux paramètres suivants :

- **Méthode de voisinage**
 - Voisinage simple (inversion de 2 produits)
 - Voisinage circulaire (rotation de 3 produits)
- **Taille du voisinage**
 - Construction complète du voisinage
 - Construction partielle (par exemple, limitation arbitraire à 20 voisins)

Il est bien évident que l'exploration de tous les voisins donnera un meilleur résultat au final, puisque le champ exploré est plus grand, mais n'est-ce pas une perte de temps ? Il s'agit de déterminer à un instant donné l'ordre de lancement des produits en entrée du système. Cette décision doit être prise en temps quasi réel. Le temps de calcul est dans ce cas un facteur de performance primordial. Par conséquent, il nous paraît plus intéressant d'analyser les performances par rapport au temps de calcul que d'analyser le seul résultat.

3. Résultats

3.1. Résultat de l'heuristique

La première analyse à effectuer consiste à regarder la répartition des ordres issus de l'heuristique en fonction de l'individu de départ. L'individu de départ étant un individu choisi aléatoirement, il en existe 369600 pour le problème 1, et 656 pour le problème 2.

La répartition du temps de production, pour chacune des instances, est reprise dans les figures :

| | | Taille de la population | |
|------------------|---------------------|-------------------------|--------------|
| | | Complète | 20 individus |
| Type d'inversion | Double | Figure 67 | Figure 68 |
| | Triple (circulaire) | Figure 69 | Figure 70 |

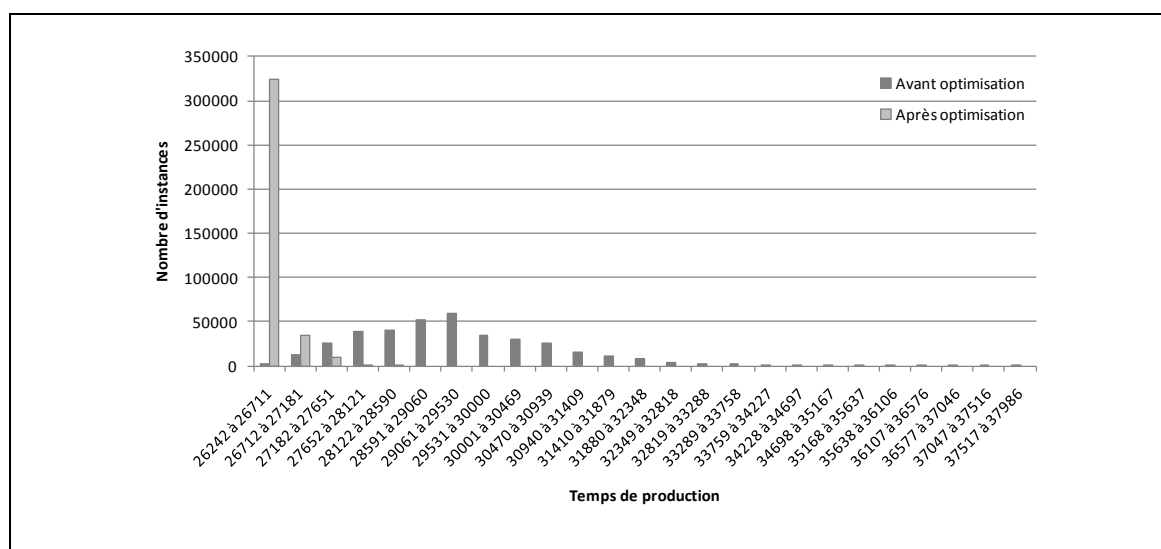


Figure 67 : Répartition après optimisation, problème 1, inversion double, 54 individus par pas

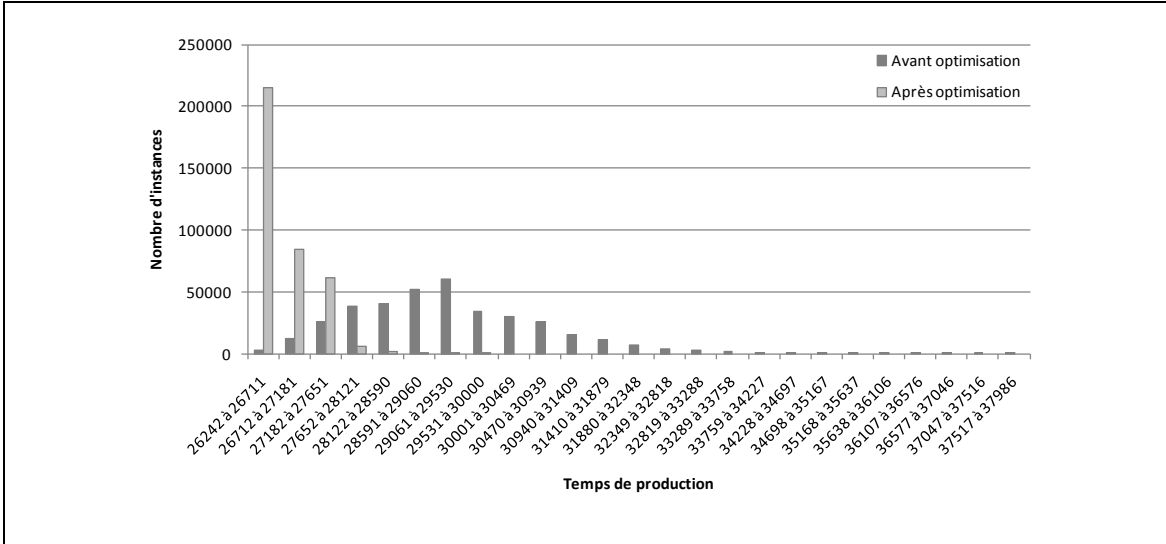


Figure 68 : Répartition après optimisation, problème 1, inversion double, 20 individus par pas

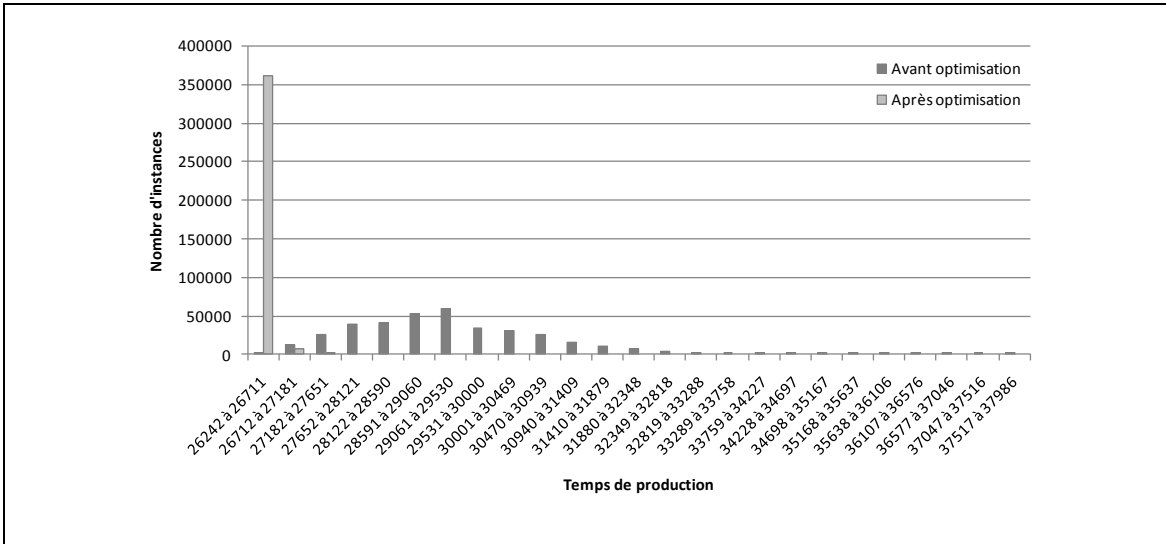


Figure 69 : Répartition après optimisation, problème 1, inversion triple, 432 individus par pas

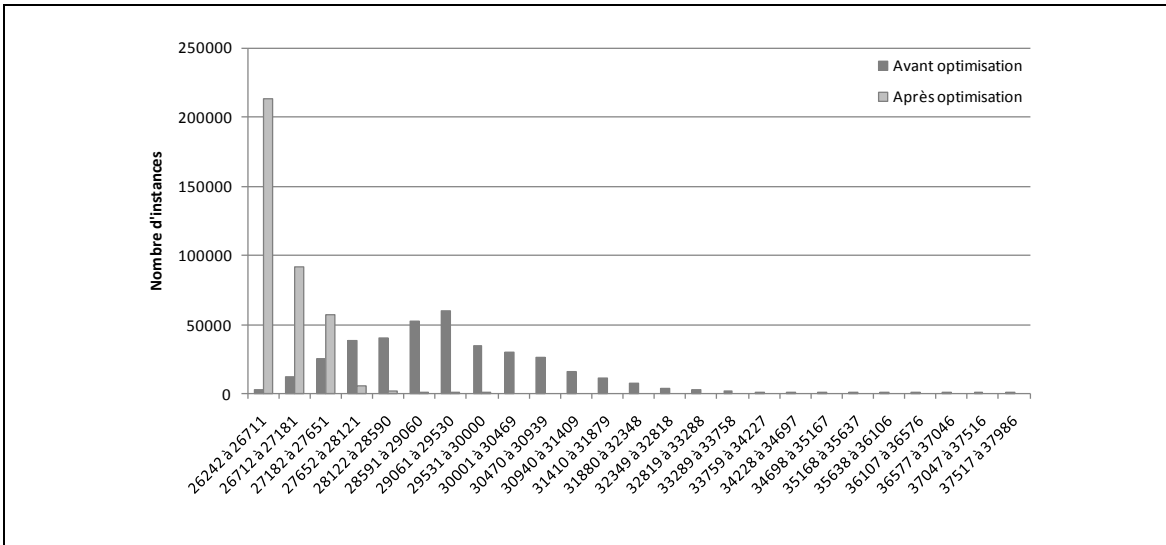


Figure 70 : Répartition après optimisation, problème 1, inversion triple, 20 individus par pas

L'objectif de l'heuristique étant de trouver l'ordre de lancement produisant le meilleur makespan, le graphique idéal serait constitué d'une unique colonne à gauche du graphique. Ainsi, plus la répartition après optimisation se rapproche de la répartition idéale, plus l'heuristique est bonne.

Si l'on se base sur **ce seul critère**, le meilleur paramétrage de l'heuristique consiste à utiliser des inversions triples (ou circulaires) en explorant tous les voisins à chaque pas d'optimisation.

Si l'on compare les **deux instances avec une limitation** du nombre de voisins par pas de simulation, on remarque que les deux résultats sont similaires. Ainsi, on peut penser à première vue que seule la taille du voisinage a un impact sur le résultat de l'heuristique.

Cependant, le temps de calcul nécessaire à la convergence de l'heuristique est lié d'une part au nombre de pas d'optimisation nécessaires à la convergence et d'autre part à la taille de la population de chaque pas. On peut penser que l'évaluation de tous les voisins, à chaque pas d'optimisation, n'est pas nécessaire. En effet on ne sélectionne qu'un voisin à chaque pas, alors que nous testons beaucoup plus d'inversions, qui seront ré-analysées dans les pas suivants sous une autre forme (les inversions locales sont quasiment indépendantes les unes des autres). La question devient alors celle-ci : « *Est-il préférable de lancer successivement plusieurs fois l'heuristique avec une taille restreinte de population, ou une seule fois avec une exploration complète du voisinage ?* »

3.2. Étude du nombre de pas d'optimisation

Nous avons vu dans l'étude précédente quel était le paramétrage optimal pour obtenir le meilleur résultat en cas de lancement d'une seule instance de l'heuristique.

Les histogrammes résultants n'étant pas complètement différents, nous proposons de comparer le nombre de pas de l'heuristique avec et sans limitation du nombre d'enfants. La répartition des instances par nombre de pas nécessaires à la convergence sont repris dans la figure 71 et la figure 72. Dans aucun cas, l'heuristique n'a eu besoin de plus de 8 pas pour converger.

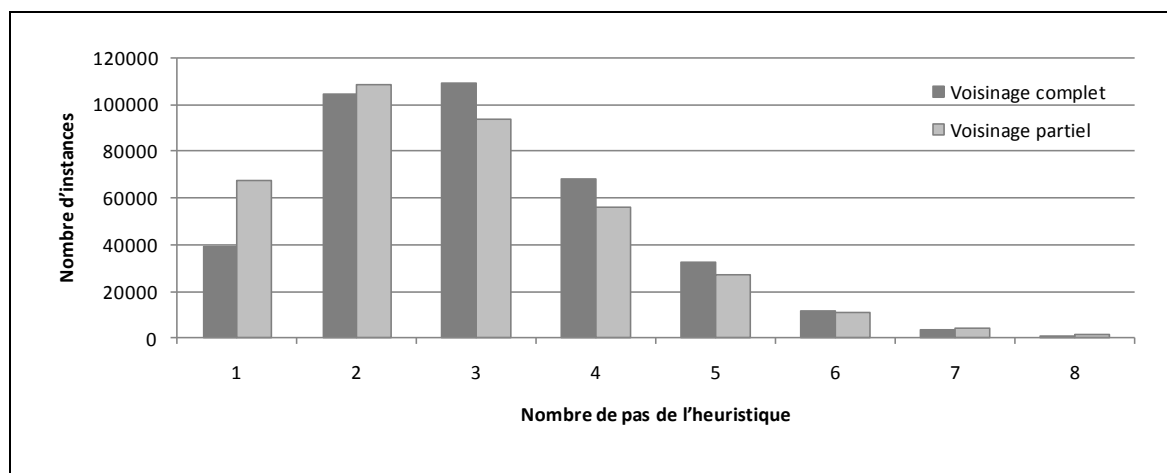


Figure 71 : Comparaison du nombre de pas de l'heuristique, problème 1, inversion double

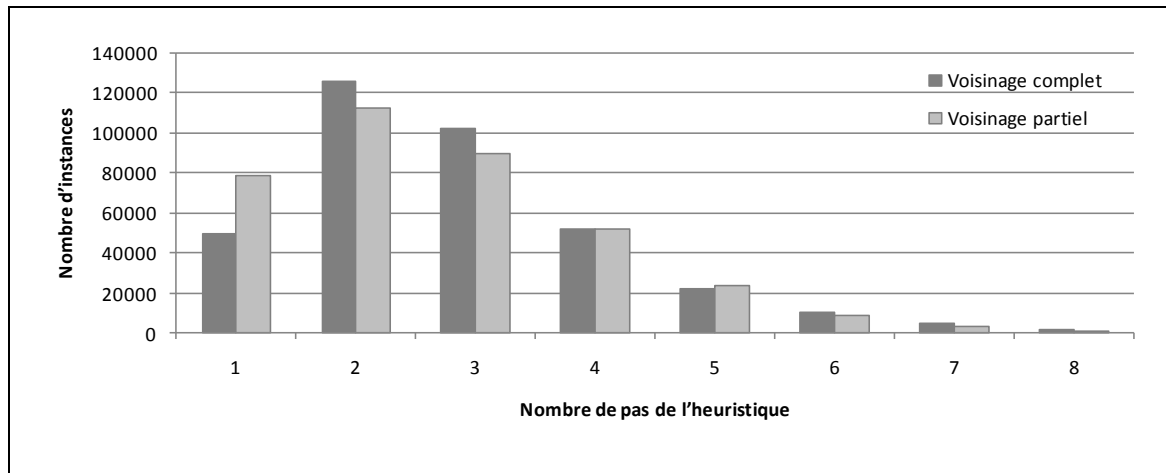


Figure 72 : Comparaison du nombre de pas de l'heuristique, problème 1, inversion triple

Dans le cas d'une inversion double et d'une inversion triple, le nombre de pas nécessaires à la convergence de l'heuristique reste le même. Une différence entre l'exploration du voisinage complet et du voisinage partiel réside alors dans :

- le nombre d'évaluations de performances à chaque pas
- le minimum issu de l'heuristique

Dans le cas d'une inversion double, l'exploration complète du voisinage demande 2.5 fois plus d'évaluation, donc 2.5 fois plus de temps de calcul que dans le cas de l'exploration partielle. Dans le cas d'une inversion triple, ce facteur est de 37. Le meilleur résultat de certaines heuristiques justifie-t-il une telle différence dans le temps de calcul ?

3.3. Étude du lancement successif de plusieurs instances de l'heuristique

Les résultats des études précédentes nous amènent à comparer le temps de convergence vers une '*solution optimale*'. Nous appelons ici '*solution optimale*' un ordre de lancement dont le temps de production est compris entre le temps de production de la solution réellement optimale et 1,05 fois ce temps. En d'autres termes, nous tolérons un temps de réalisation des produits situé à 5% de l'optimum global.

Pour comparer les différentes performances, nous proposons de représenter sur un graphique l'évolution de la probabilité de trouver une solution optimale (notée p_n), et le temps de calcul (noté t_n) pour y parvenir.

- premier pas de l'heuristique :
 - le temps de calcul t_1 est lié au temps de lancement de la simulation et au nombre de simulations à réaliser
 - la probabilité p_1 d'obtenir un résultat à 5% de l'optimum est calculée à partir de l'analyse exhaustive

- pas suivants :

- le temps de calcul est calculé comme suit :

$$t_n = n * t_1$$

- la probabilité d'avoir une solution à 5% de l'optimum est :

$$p_n = 1 - (1 - p_1)^n$$

En transformant ces valeurs discrètes en courbe continue, on a le modèle :

$$p(t) = 1 - (1 - p_1)^{\frac{t}{t_1}}$$

La représentation de ces courbes sur un graphique temps/probabilité est donnée dans la figure 73.

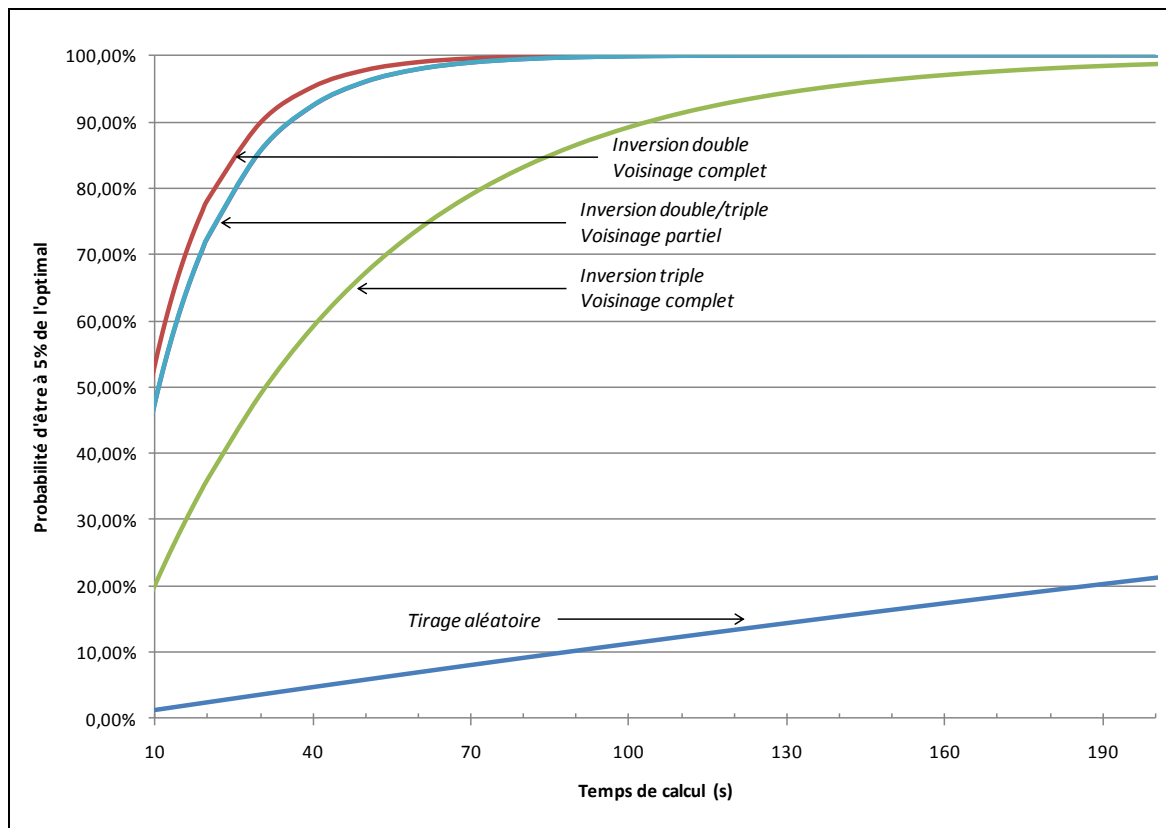


Figure 73 : Performances temporelles de 4 instances de l'heuristique, problème 1

Cette analyse comparative montre que le paramétrage de l'heuristique pour obtenir un grand voisinage n'est pas la meilleure solution au final. En effet, les calculs à chaque pas sont très longs, ce qui induit de mauvaises performances dans le temps.

Les trois autres instances ont des performances temporelles comparables. Cependant, l'optimisation devant s'arrêter à la fin d'une des instances de l'heuristique, il est préférable de choisir l'instance proposant le plus de points d'arrêt, donc celle où l'heuristique a le plus faible temps d'exécution. Il faudra donc choisir une inversion à voisinage partiel.

4. Conclusion

Ce chapitre a démontré, dans le cas d'un problème de traitement de surface, et probablement dans tout système réactif, que le lancement des ordres de fabrication dans une unité de production peut avoir un fort impact sur les performances de celle-ci. L'auteur, dans [Cardin O., 2007] avait proposé une étude similaire sur l'étude du regroupement des ordres de fabrication. L'étude avait abouti aux mêmes conclusions : une étude avant le lancement de la production est bien souvent utile et l'impact sur les performances de l'unité peut être important.

Dans le cas où plusieurs ordres de production sont disponibles et que leurs dates de livraisons le permettent, ils nous semblent important d'étudier finement l'ordre de lancement afin d'obtenir la meilleure productivité de l'installation. Ce chapitre expose une heuristique de génération d'un ordre optimal dans le cadre du pilotage réactif des installations de traitement de surface.

5. Annexes : résultats du problème 2

Note : Dans le cas du problème 2, le voisinage de chaque solution étant déjà de faible cardinal, une limitation supplémentaire serait inutile.

| | Problème 1 | Problème 2 |
|---|------------|------------|
| Répartition des instances par temps de production | Figure 64 | Figure 74 |
| Répartition après optimisation, inversion double, voisinage complet | Figure 67 | Figure 75 |
| Répartition après optimisation, inversion double, voisinage partiel | Figure 68 | |
| Répartition après optimisation, inversion triple, voisinage complet | Figure 69 | Figure 76 |
| Répartition après optimisation, inversion triple, voisinage partiel | Figure 70 | |
| Comparaison du nombre de pas de l'heuristique, inversion double | Figure 71 | Figure 77 |
| Comparaison du nombre de pas de l'heuristique, inversion triple | Figure 72 | Figure 78 |
| Performances temporelles de l'heuristique | Figure 73 | Figure 79 |

Tableau 11 : Correspondance entre les résultats des problèmes 1 et 2

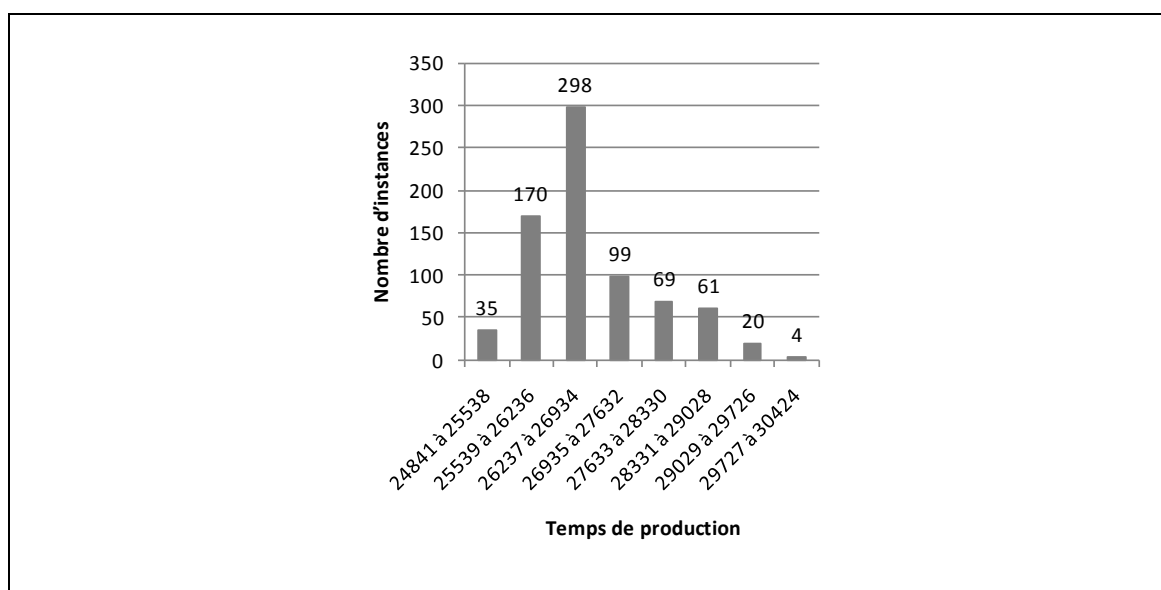


Figure 74 : Répartition des 656 instances du problème 2 par temps de production

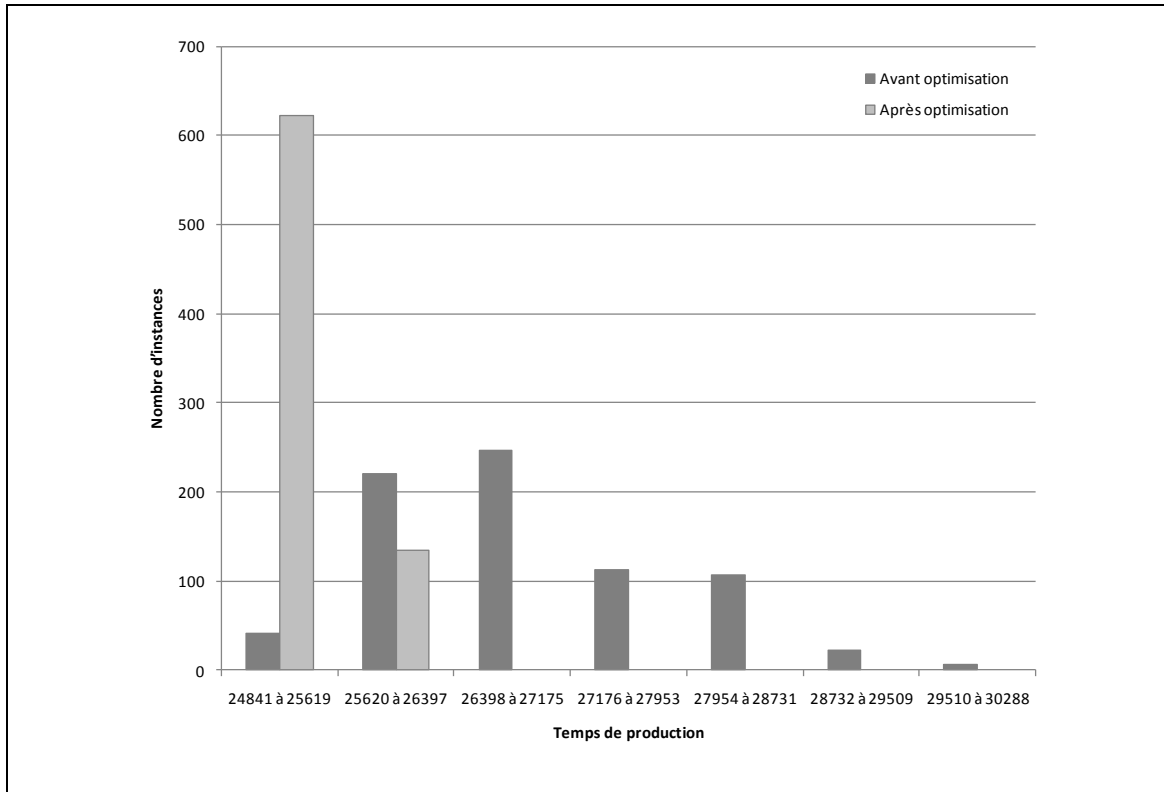


Figure 75 : Répartition après optimisation, problème 2, inversion double, voisinage complet

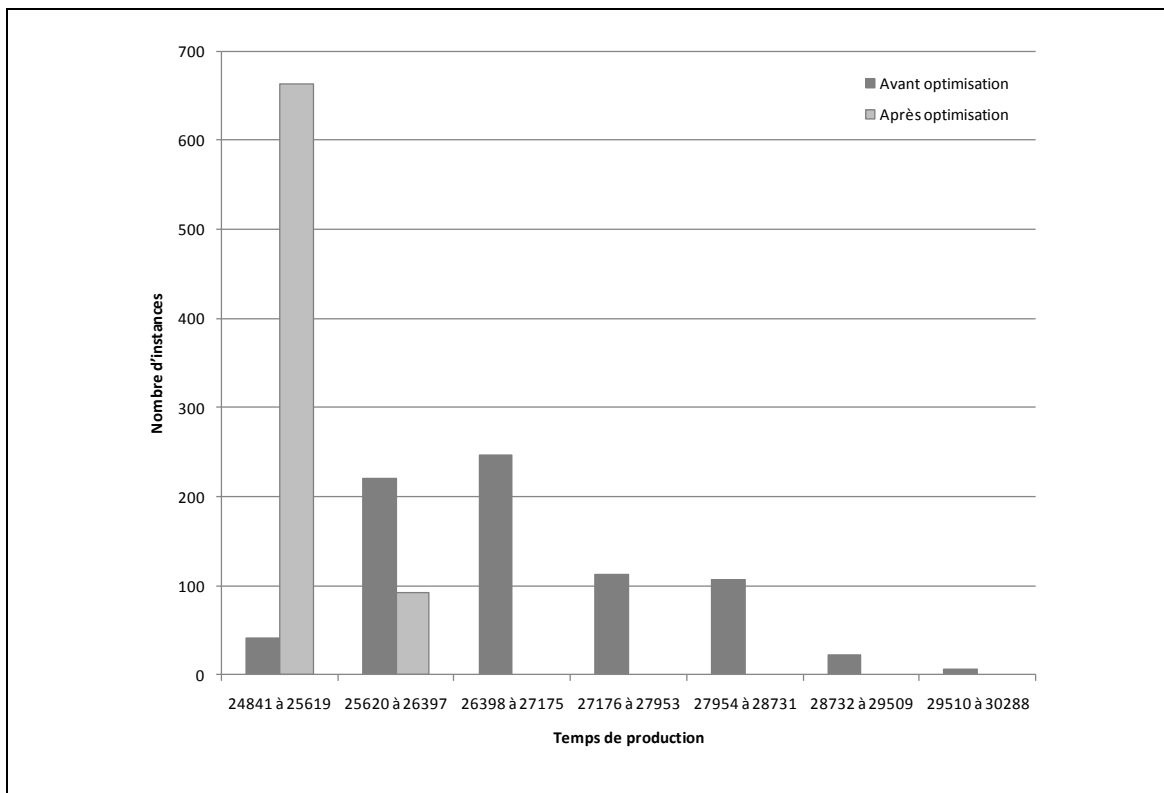


Figure 76 : Répartition après optimisation, problème 2, inversion triple, voisinage complet

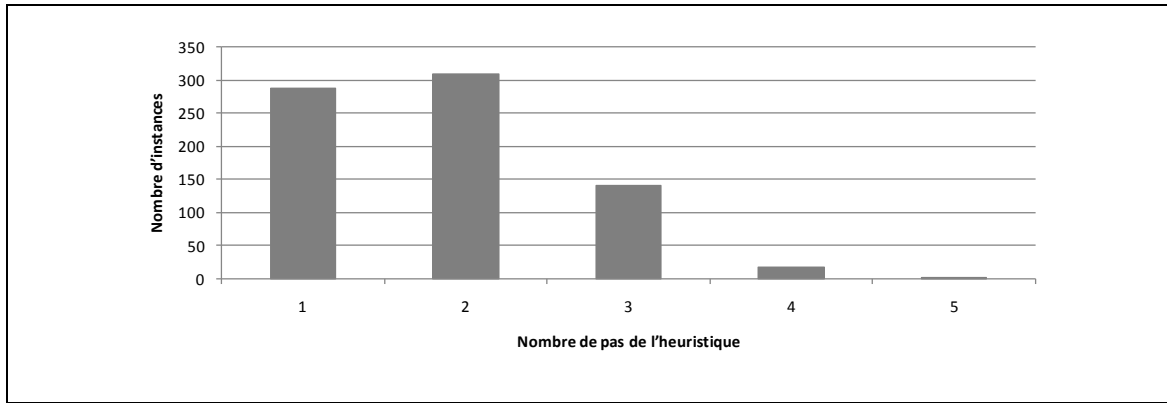


Figure 77 : Nombre de pas de l'heuristique, problème 2, inversion double, voisinage partiel

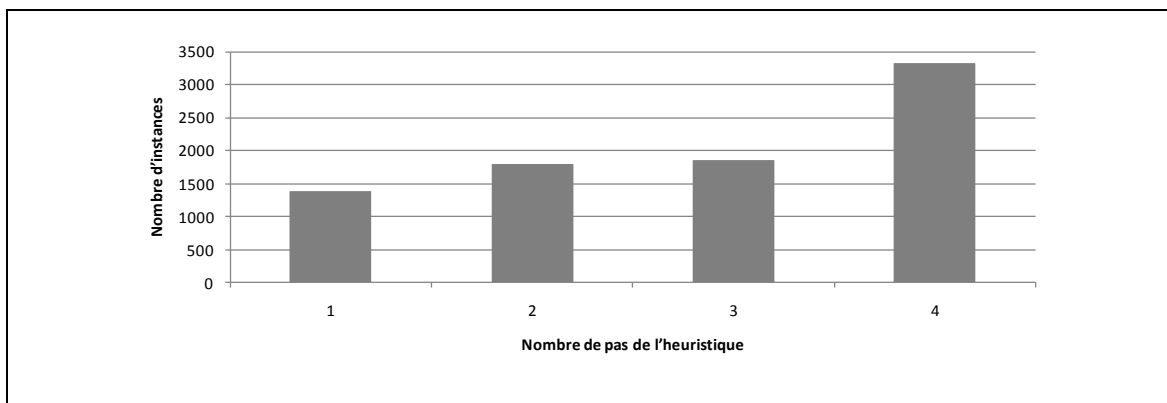


Figure 78 : Nombre de pas de l'heuristique, problème 2, inversion triple, voisinage partiel

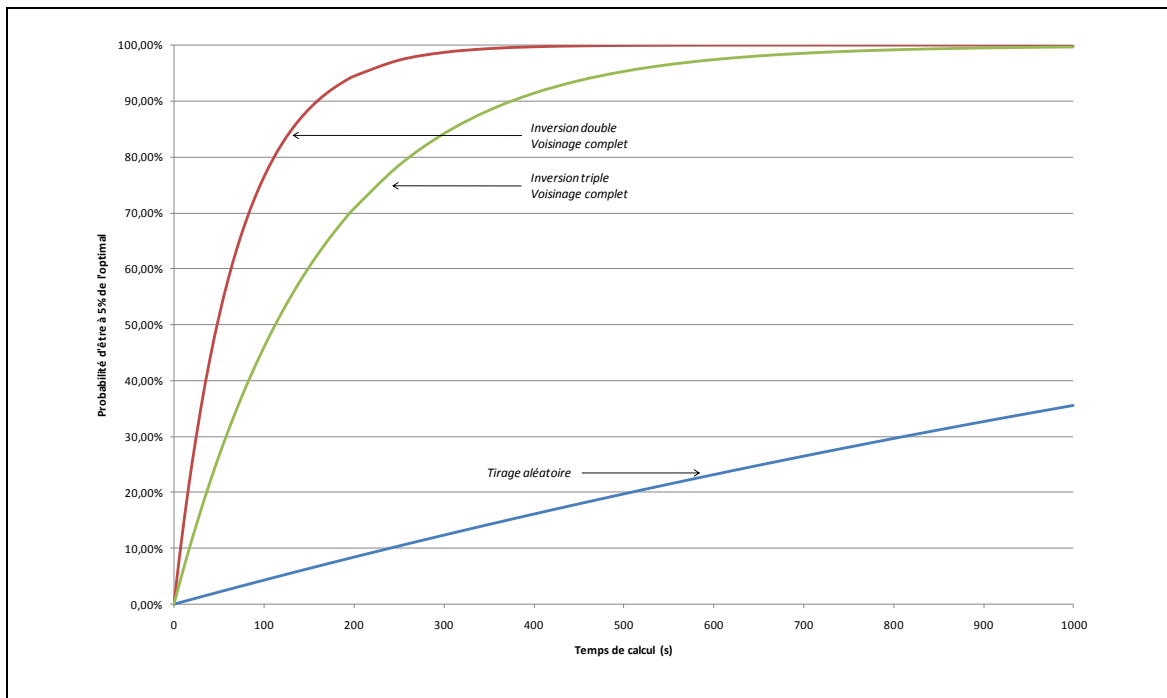


Figure 79 : Performances temporelles de 2 instances de l'heuristique, problème 2

Conclusion et perspectives

Conclusion

L'objectif de cette thèse était de répondre à une double problématique, industrielle et scientifique, autour du pilotage des installations de traitement de surface.

La problématique industrielle s'intéresse, dans le contexte fortement concurrentiel actuel, à augmenter la productivité et la flexibilité d'une installation de production, tout en assurant les contraintes temporelles fortes liées au procédé de fabrication. Cette problématique nous a été confiée, par une entreprise du secteur aéronautique français, dans le cadre de plusieurs contrats industrie-recherche.

La problématique scientifique proposée, est d'appliquer le concept du pilotage réactif de la production (*particulièrement étudié au sein du groupe Systèmes Contrôlés par le Produit – SCP – devenu cette entre temps Intelligent Manufacturing and Services Systems – IMS² – du GDR MACS*), au pilotage des installations de traitement de surface. Ce concept présente l'avantage d'assurer une grande flexibilité et une bonne réactivité (voir chapitre 1). Il est cependant contradictoire avec les contraintes temporelles fortes du problème industriel.

Compte tenu des fortes contraintes temporelles énoncées dans le chapitre 2, nous avons montré (dans le chapitre 3) qu'il était impossible de piloter une installation de ce type uniquement avec un mode de pilotage réactif. Nous avons alors proposé, un couplage entre un pilotage réactif et un pilotage prédictif. La partie réactive assure une réponse rapide aux aléas, tandis que la partie prédictive garantit les contraintes temporelles fortes du problème.

Cette thèse relative au couplage de deux modes de pilotage dans le cadre de l'ordonnancement des installations de traitement de surface a été validée par simulation puis implantée au niveau industriel dans une unité de production (Aerolia, Saint-Nazaire).

Les chapitres 4 et 5 ont ouvert la réflexion autour de ce mode de pilotage, clôturant ainsi les trois optimisations d'un système de production évoquées à la fin du chapitre 2. Le chapitre 4 a exposé l'application de la méta-heuristique génétique à la construction automatique d'une topologie d'atelier de traitement de surface. Les comparaisons des résultats avec une topologie d'expert a permis de valider la méthode de construction.

Enfin, le dernier chapitre de cette thèse a souligné l'importance, dans un système réactif, d'étudier l'ordre de lancement des produits. Cette étude, basée sur un algorithme de descente simple, montre l'impact du ré-ordonnancement des *commandes clients* sur la productivité d'une installation.

Perspectives

Au niveau scientifique, nous avons proposé un couplage entre deux modes de pilotage sur un problème bien particulier. Ce couplage pourrait être appliqué à d'autres systèmes de production tels

que les chaînes flexibles de production, le stockage dynamique, le pilotage de chariots autoguidés... autant de domaines étudiés dans l'équipe *ACSED* du laboratoire *IRCCyN*. La première perspective serait donc d'étudier l'application de ce mode de pilotage sur d'autres problèmes munis de contraintes spatiales et temporelles différentes.

Comme nous l'avons vu, un système de pilotage réactif comporte de nombreux paramètres de réglage pouvant, à un instant donné, être réglés à travers un plan d'expérience. Le problème, est que ce bon réglage dépend d'une part des gammes de production et d'autre part des combinaisons de produits lancées. Or ces deux choses sont susceptibles d'évoluer au cours du temps. Une des perspectives d'évolution serait d'imaginer un module capable, à partir de l'observation des performances du système, de faire évoluer automatiquement certains de ces paramètres.

De plus, une des perspectives de cette thèse est l'application au niveau industriel de ces travaux. La première étude présentée dans le chapitre 3 a déjà fait l'objet de nombreux contrats industrie-recherche avec Aerolia Saint-Nazaire, industriel de l'aéronautique, débouchant sur l'implantation des algorithmes dans un atelier de traitement de surface. Il a été évoqué, au début de cette étude avec l'industriel, l'utilité des deux études suivantes (chapitre 4 et 5). Une des perspectives envisagées serait alors le déploiement des algorithmes dans l'installation de production industrielle.

La perspective de l'application du chapitre 5 soulève alors une problématique scientifique de la nécessité, à chaque instant, de connaître l'état du système de production. Lorsque toutes les grandeurs ne sont pas mesurables, une problématique concernant la reconstruction des informations manquantes doit être résolue. Ce développement se basera sur les travaux de [Cardin O. *et al.*, 2005] et [Cardin O. *et al.*, 2008] concernant la reconstruction par simulation de l'état d'un système de production.

Liste des figures

| | |
|--|----|
| Figure 1 : Exemple de pyramide CIM d'un site de production | 10 |
| Figure 2 : Pyramide CIM appliquée au niveau de l'entreprise | 11 |
| Figure 3 : Les deux mécanismes d'agencement : hiérarchie et hétérarchie | 12 |
| Figure 4 : Organisation d'une holoarchie | 13 |
| Figure 5 : Flat Holonic Form | 14 |
| Figure 6 : Évolution des systèmes organisés | 14 |
| Figure 7 : Exemple de FMS proposé dans le brevet USP 4.472.783..... | 15 |
| Figure 8 : Organisation d'un système reconfigurable | 16 |
| Figure 9 : Paradigmes de productions classés par objectif | 17 |
| Figure 10 : Sources de perturbations en production selon [Cauvin A., 2005] | 18 |
| Figure 11 : Cycle de vie d'une perturbation selon [Cardin O., 2007] | 19 |
| Figure 12 : Pilotage dynamique d'un système | 21 |
| Figure 13 : Architecture d'évaluation d'un ré-ordonnanceur | 22 |
| Figure 14 : Atelier de production à deux machines | 23 |
| Figure 15 : Lien entre règles locales et comportement global | 24 |
| Figure 16 : Principe de fonctionnement des trois classes de pilotage | 25 |
| Figure 17 : Comparaison des trois classes de pilotage..... | 26 |
| Figure 18 : Exemple de ligne de traitement de surface | 28 |
| Figure 19 : Exemple de cas de blocage circulaire..... | 30 |
| Figure 20 : Cuve multi emplacements ou cuves dupliquées | 31 |
| Figure 21 : Décomposition d'une mission de pont..... | 32 |
| Figure 22 : Représentation d'un cycle robot sur un line-graph..... | 33 |
| Figure 23 : Ligne de traitement de surface multi-robot..... | 34 |
| Figure 24 : Installation de traitement de surface avec navette | 34 |
| Figure 25 : Implantation type d'une unité de traitement de surface..... | 35 |
| Figure 26 : Exemple d'ordonnancement cyclique d'une unité à trois cuves..... | 46 |

| | |
|--|----|
| Figure 27 : Les deux étapes de [Bhushan S. <i>et al.</i> , 2003] | 48 |
| Figure 28 : Les trois optimisations d'un système de production..... | 54 |
| Figure 29 : Exemple d'implantations abordées | 57 |
| Figure 30 : Installation industrielle A..... | 59 |
| Figure 31 : Installation industrielle B..... | 59 |
| Figure 32 : Architecture client/serveur appliquée au RHSP | 60 |
| Figure 33 : Transfert d'une pièce à travers une zone commune..... | 64 |
| Figure 34 : Bibliothèque Siman/Arena : Pont..... | 66 |
| Figure 35 : Bibliothèque Siman/Arena : Bain | 66 |
| Figure 36 : Bibliothèque Siman/Arena : Attente | 67 |
| Figure 37 : Bibliothèque Siman/Arena : Lancement | 67 |
| Figure 38 : Bibliothèque Siman/Arena : Navette | 67 |
| Figure 39 : Modèle logique du bain sous Siman/Arena | 69 |
| Figure 40 : Bibliothèque Siman/Arena : Modélisation de l'installation industrielle A | 70 |
| Figure 41 : Exemple de visualisation graphique | 72 |
| Figure 42 : Influence de la limite de barres sur le temps de production (4 attentes par ligne) | 74 |
| Figure 43 : Influence de la limite de barres sur le nombre de dépassements critiques | 75 |
| Figure 44 : Couche prédictive de la sélection des missions du pont..... | 77 |
| Figure 45 : Diagramme UML du pilotage réactif | 78 |
| Figure 46 : Évolution des performances avec le planning prédictif | 79 |
| Figure 47 : Évaluation d'un système de pilotage..... | 80 |
| Figure 48 : Exemple d'implémentation dynamique et réactive | 81 |
| Figure 49 : Phases d'implantation dans le cycle de vie d'un atelier de production | 86 |
| Figure 50 : Typologie des emplacements à implanter | 87 |
| Figure 51 : Dynamique d'un algorithme génétique selon [Forrest S., 1996] | 90 |
| Figure 52 : Génération d'un individu de la population de départ..... | 91 |
| Figure 53 : Mécanisme de reproduction/sélection | 92 |
| Figure 54 : Mécanisme de croisement | 93 |
| Figure 55 : Blocs élémentaires à implanter pour le cas industriel B | 95 |

| | |
|--|-----|
| Figure 56 : Implémentation de l'algorithme génétique | 96 |
| Figure 57 : Évolution du meilleur individu de chaque génération (unité arbitraire) | 97 |
| Figure 58 : Comparaison des différentes topologies | 98 |
| Figure 59 : Exemple de lancement de 2 produits..... | 100 |
| Figure 60 : Exemple d'ordonnement de 2 produits..... | 101 |
| Figure 61 : Architecture réseau de la répartition des simulations | 102 |
| Figure 62 : Protocole d'auto-configuration des esclaves basé sur un flux UDP/IP multicast..... | 103 |
| Figure 63 : Protocole d'exécution d'une simulation sur un esclave..... | 104 |
| Figure 64 : Répartition des 369 600 instances du problème 1 par temps de production | 105 |
| Figure 65 : Heuristique de descente..... | 106 |
| Figure 66 : Exemple de voisinage d'une séquence de lancement..... | 107 |
| Figure 67 : Répartition après optimisation, problème 1, inversion double, 54 individus par pas | 108 |
| Figure 68 : Répartition après optimisation, problème 1, inversion double, 20 individus par pas | 109 |
| Figure 69 : Répartition après optimisation, problème 1, inversion triple, 432 individus par pas..... | 109 |
| Figure 70 : Répartition après optimisation, problème 1, inversion triple, 20 individus par pas..... | 109 |
| Figure 71 : Comparaison du nombre de pas de l'heuristique, problème 1, inversion double..... | 110 |
| Figure 72 : Comparaison du nombre de pas de l'heuristique, problème 1, inversion triple | 111 |
| Figure 73 : Performances temporelles de 4 instances de l'heuristique, problème 1 | 112 |
| Figure 74 : Répartition des 656 instances du problème 2 par temps de production | 114 |
| Figure 75 : Répartition après optimisation, problème 2, inversion double, voisinage complet..... | 115 |
| Figure 76 : Répartition après optimisation, problème 2, inversion triple, voisinage complet | 115 |
| Figure 77 : Nombre de pas de l'heuristique, problème 2, inversion double, voisinage partiel | 116 |
| Figure 78 : Nombre de pas de l'heuristique, problème 2, inversion triple, voisinage partiel..... | 116 |
| Figure 79 : Performances temporelles de 2 instances de l'heuristique, problème 2 | 116 |

Liste des tableaux

| | |
|---|-----|
| Tableau 1 : Exemple de gamme de traitement | 36 |
| Tableau 2 : Tableau des contraintes d'ordonnancement..... | 38 |
| Tableau 3 : Terminologie du HSP selon [Rossé-Bloch C., 1999] | 41 |
| Tableau 4 : Choix du champ α de [Manier M.-A. <i>et al.</i> , 2003] | 43 |
| Tableau 5 : Description du champ β de [Manier M.-A. <i>et al.</i> , 2003] | 44 |
| Tableau 6 : Description du champ γ de [Manier M.-A. <i>et al.</i> , 2003]..... | 45 |
| Tableau 7 : Typologies des implantations abordées | 56 |
| Tableau 8 : Transfert d'une pièce à travers une zone commune | 65 |
| Tableau 9 : Comparaison des dépassements en fonction de la stratégie de ré-ordonnancement | 82 |
| Tableau 10 : Répartition de la production dans les deux exemples étudiés..... | 102 |
| Tableau 11 : Correspondance entre les résultats des problèmes 1 et 2 | 114 |

Bibliographie

- [Baptiste P. et al., 1992] Baptiste P., Legeard B., Varnier C., *Hoist Scheduling Problem: an Approach Based on Constraints Logic Programming*, International Conference on Robotics and Automation, Nice, 1992
- [Baptiste P. et al., 1996] Baptiste P., Legeard B., Manier M.-A., *Résolution d'un problème d'ordonnancement avec la PLC*, Journal Européen des Systèmes Automatisés, Volume 30, Issue 2, 1996, pp. 201-230
- [Berruet P., 2007] Berruet P., *Ingénierie de la commande et analyse des systèmes reconfigurables*, Habilitation à Diriger des Recherches, Université de Bretagne Sud, 2007
- [Bhushan S. et al., 2003] Bhushan S., Karimi I.-A., *An MILP approach to automated wet-etch station scheduling*, Industrial Engineering and Chemical Research, Volume 42, Issue 7, 2003, pp. 1391-1399
- [Bhushan S. et al., 2004] Bhushan S., Karimi I.-A., *Heuristics algorithms for scheduling an automated wet-etch station*, Computers and Chemical Engineering, Volume 28, 2004, pp. 363-379
- [Bi Z.-M. et al., 2008] Bi Z.-M., Lang S.-Y.-T., Shen M., Wang L., *Reconfigurable manufacturing systems: the state of the art*, International Journal of Production Research, Volume 46, Issue 4, 2008, pp. 967-992
- [Blanc P., 2006] Blanc P., *Pilotage par approche holonique d'un système de production de vitres de sécurité feuilletées*, Thèse de doctorat, École Centrale de Nantes et Université de Nantes, 2006
- [Blazewicz J. et al., 1993] Blazewicz J., Eker K.-H., Schmidt G., Weglarz J., *Scheduling in Computer and Manufacturing Systems*, Springer-Verlag, Berlin, 1993
- [Bloch C. et al., 1999] Bloch C., Manier M.-A., *Notation and Typology for the Hoist Scheduling Problem*, International Conference on Systems, Man and Cybernetics, Tokyo, Japan, 1999, pp. 475-480
- [Bongaerts L. et al., 2000] Bongaerts L., Monostori L., McFarlane D., Kadar B., *Hierarchy in distributed shop floor control*, Computers in Industry, Volume 43, 2000, pp. 123-137

- [Brandimarte P. et al., 1995] Brandimarte P., Cantamessa M., *Methodologies for designing CIM systems: A critique*, Computers in Industry, Issue 25, 1995, pp. 181-193
- [Breton L. et al., 2004] Breton L., Maza S., Castagna P., *Simulation multi-agent de systèmes d'AGVs : Comparaison avec une approche prédictive*, MODélisation et SIMulation, Nantes, France, 2004
- [Brucker P., 2006] Brucker P., *Scheduling Algorithms*, 5, Osnabrück, Springer, 2006
- [C Fu M., 1992] C Fu M., *Optimisation via simulation: A review*, Annals of Operations Research, Volume 53, 1992, pp. 199-248
- [Cardin O. et al., 2005] Cardin O., Castagna P., *Defining a command architecture enabling proactive simulation on a complex manufacturing system*, Conceptual Modeling and Simulation Conference, Marseille, France, 2005, pp. 205-210
- [Cardin O. et al., 2008] Cardin O., Castagna P., *Proactive production activity control by online simulation*, International Journal of Simulation and Process Modeling, 2008
- [Cardin O. et al., 2008a] Cardin O., Castagna P., Chové E., *Contribution of simulation in the product-driven systems production activity control*, IFAC, Séoul, Corée, 2008
- [Cardin O. et al., 2008b] Cardin O., Castagna P., Chové E., *La simulation pour l'aide au pilotage des systèmes contrôlés par le produit*, MODélisation et SIMulation, Paris, 2008
- [Cardin O. et al., 2009] Cardin O., Castagna P., Chové E., *Utilisation d'un observateur pour le pilotage des systèmes contrôlés par le produit*, JESA, Volume 43, Issue 4-5, 2009
- [Cardin O., 2007] Cardin O., *Apport de la simulation en ligne dans l'aide à la décision pour le pilotage des systèmes de production - Application à un système flexible de production*, Thèse de doctorat, Université de Nantes, 2007
- [Cauvin A., 2005] Cauvin A., *Analyse, modélisation et amélioration de la réactivité des systèmes de décision dans les organisations industrielles*, Habilitation à Diriger des Recherches, Université Paul Cézanne - Aix-Marseille III, 2005
- [Cea Ramirez A.-A., 2006] Cea Ramirez A.-A., *Contribution à la Modélisation et à la Gestion des Interactions Produit-Processus dans la Chaîne Logistique par l'Approche Produits Communicants*, Thèse de doctorat, Université Henri Poincaré, 2006

- [Chauvet F. *et al.*, 2000] Chauvet F., Levner E., Meyzin L. K., Proth J.-M., *On-line scheduling in a surface treatment system*, European Journal of Operational Research, Volume 120, Issue 2, 2000, pp. 382-392
- [Che A. *et al.*, 2005] Che A., Chu C., *A polynomial algorithm for no-wait cyclic hoist scheduling in an extended electroplating line*, Operations Research Letters, Volume 33, 2005, pp. 274-284
- [Che A. *et al.*, 2007] Che A., Chu C., *Cyclic hoist scheduling in large real-life electroplating lines*, OR Spectrum, Volume 29, Issue 3, 2007, pp. 445-470
- [Chové E. *et al.*, 2009a] Chové E., Castagna P., Abbou R., *Utilisation conjointe d'approches réactives et prédictives : Application à une ligne de traitement de surface*, Conception et Production Intégrées, Fès, Maroc, 2009
- [Chové E. *et al.*, 2009b] Chové E., Castagna P., Abbou R., *Hoist Scheduling Problem: Coupling reactive and predictive approaches*, INCOM, Moscou, Russie, 2009
- [Chové E. *et al.*, 2010] Chové E., Castagna P., Abbou R., *Simulation contribution for hoist scheduling problem and hoist plant topology design*, International Conference of MOdeling and SIMulation, Hammamet, Tunisia, 2010
- [Collinot A. *et al.*, 1996] Collinot A., Droguoul A., *La méthode de conception multi-agent CASSIOPEE : Application à la robotique collective*, Technical report LAFORIA, LIP6, 1996
- [Dauzère-Pérès S. *et al.*, 1998] Dauzère-Pérès S., Roux W., Lasserre J.-B., *Multi-resource shop scheduling with resource flexibility*, European Journal of Operational Research, Volume 107, 1998, pp. 289-305
- [Dorn J. *et al.*, 1995] Dorn J., Kerr R., Thalhammer G., *Reactive scheduling: improving the robustness of schedules and restricting the effects of shop floor disturbances by fuzzy reasoning*, International Journal of Human-Computer Studies, Volume 42, Issue 6, 1995, pp. 687-704
- [Drobouchevitch I.-G. *et al.*, 2006] Drobouchevitch I.-G., Sethi S.-P., Sriskandarajah C., *Scheduling dual gripper robotic cell: One-unit cycles*, European Journal of Operational Research, Volume 171, 2006, pp. 598-631
- [Fargier H. *et al.*, 2001] Fargier H., Lamothe J., *Handling soft constraints in hoist scheduling problems: the fuzzy approach*, Engineering Applications of Artificial Intelligence, Volume 14, 2001, p. 387-399

- [Fleury G. *et al.*, 1999] Fleury G., Gourgand M., Lacomme P., *Coupling meta-heuristics and discrete event simulation models for the stochastic hoist scheduling problem*, ACS99, 1999
- [Fleury G. *et al.*, 2001] Fleury G., Gourgand M., Lacomme P., *Meta-heuristics for the stochastic hoist scheduling problem*, International Journal of Production Research, Volume 39, Issue 15, 2001, pp. 3419-3457
- [Forrest S., 1996] Forrest S., *Genetic Algorithms*, ACM Computing Surveys, Volume 28, Issue 1, 1996, pp. 77-80
- [Geiger C.-D. *et al.*, 1997] Geiger C.-D., Kempf K.-G., Uzsoy R., *A tabu search approach to scheduling an automated wet etch station*, Journal of Manufacturing Systems, Volume 16, Issue 2, 1997, pp. 102-116
- [Gourgand M. *et al.*, 2003] Gourgand M., Grangeon N., Norre S., *A contribution to the stochastic flow shop scheduling problem*, European Journal of Operational Research, Volume 151, 2003, pp. 415-433
- [Gourgand M. *et al.*, 2005] Gourgand M., Grangeon N., Norre S., *Markovian analysis for performance evaluation and scheduling in m machine stochastic flow-shop with buffers of any capacity*, European Journal of Operational Research, Volume 161, 2005, pp. 126-147
- [Graham R.-L. *et al.*, 1979] Graham R.-L., Lawler E.-L., Lenstra J.-K., Rinnoy Kan A.-H.-G., *Optimisation and Approximation in Deterministic Sequencing and Scheduling Theory: A Survey*, Annals of Discrete Mathematics, Volume 5, 1979, pp. 287-326
- [Herroelen W. *et al.*, 2004] Herroelen W., Leus R., *Robust and reactive project scheduling : a review and classification of procedures*, International Journal of Production Research, Volume 42, Issue 8, 2004, pp. 1599-1620
- [Hindi K.-S. *et al.*, 2004] Hindi K.-S., Fleszar K., *A constraint propagation heuristic for the single-hoist, multiple-products scheduling problem*, Computer and Industrial Engineering, Volume 47, 2004, pp. 91-101
- [Huyet A.-L. *et al.*, 2004] Huyet A.-L., Paris J.-L., *Extraction de connaissances pertinentes sur le comportement des systèmes de production : une approche conjointe par Optimisation Évolutionniste via Simulation et Apprentissage*, Journées du GDR MACS, LIMOS, 2004

- [IEC 61512, 1997] IEC 61512, *Batch control*, International Electrotechnical Commission, 1997
- [IEC 7498-1, 1994] IEC 7498-1, *Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*, International Electrotechnical Commission, 1994
- [Johnstone R. et al., 1984] Johnstone R., Kurtzhaltz J. E., *Flexible Manufacturing System*, Brevet 4.472.783, United States Patent, 1984
- [Kats V. et al., 2008] Kats V., Lei L., Levner E., *Minimizing the cycle time of multiple-product processing networks with a fixed operation sequence, setups, and time-window constraints*, European Journal of Operational Research, Volume 187, Issue 3, 2008, pp. 1196-1211
- [Koestler A., 1969] Koestler A., *The Ghost in the Machine*, Londres, Arkana Books, 1969
- [Lamothe J. et al., 1995] Lamothe J., Corregé M., Delmas J., *A dynamic heuristic for the real time hoist scheduling problem*, IEEE symposium on emerging technologies and factories automation, Paris, 1995, IEEE Computer Society Press, pp. 161-168
- [Lemoine D. et al., 2010] Lemoine D., Bellenguez-Morineau O., Guéret C., Najid N., *Résolution d'un problème réel de type job-shop flexible multi-ressources*, ROADEF, Toulouse, 2010
- [Leung J.-M.-Y. et al., 2006] Leung J.-M.-Y., Levner E., *An efficient algorithm for multi-hoist cyclic scheduling with fixed*, Operations Research Letters, Volume 34, 2006, pp. 465-472
- [Levner E. et al., 2010] Levner E., Kats V., Alcaide López de Pablo D., Cheng T.-C.-E., *Complexity of cyclic scheduling problems: A state-of-the-art survey*, Computers and Industrial Engineering, 2010
- [Liebermana R.-W. et al., 1981] Liebermana R.-W., Turksen I.-B., *Crane Scheduling Problems*, IIE Transactions, Volume 13, Issue 4, 1981, pp. 304-311
- [Mangione F., 2003] Mangione F., *Ordonnancement des ateliers de traitement de surface pour une production cyclique et mono-produit*, Thèse de doctorat, Institut National Polytechnique de Grenoble, 2003

- [Manier M.-A. et al., 2000] Manier M.-A., Varnier C., Baptiste P., *Constraint-based model for the cyclic multi-hoists scheduling problem*, Production Planning and Control, Volume 11, Issue 3, 2000, pp. 244-257
- [Manier M.-A. et al., 2003] Manier M.-A., Bloch C., *A Classification for Hoist Scheduling Problems*, International Journal of Flexible Manufacturing Systems, Volume 15, 2003, pp. 37-55
- [Manier M.-A., 1994] Manier M.-A., *Contribution à l'ordonnancement cyclique du système de manutention d'une ligne de galvanoplastie*, Thèse de doctorat, Université de Besançon, 1994
- [Mehrabi M.-G. et al., 2000] Mehrabi M.-G., Ulsoy A.-G., Koren Y., *Reconfigurable manufacturing system: Key to future manufacturing*, Journal of Intelligent Manufacturing, Volume 11, 2000, pp. 403-419
- [Mori M. et al., 2009] Mori M., Fujishima M., *Changeable and Reconfigurable Manufacturing Systems*, 1ère édition, London, Springer-Verlag, 2009
- [Müller J.-P. et al., 2005] Müller J.-P., Ratzé C., Gillet F., Stoffel K., *Modeling and simulating hierarchies using an agent-based approach*, International Congress on Modelling and Simulation, Melbourne, 2005, pp. 1631-1638
- [Ng W.-C., 1995] Ng W.-C., *Determining the optimal number of duplicated process tanks in a single-hoist circuit board production line*, Computers and Industrial Engineering, Volume 28, Issue 4, 1995, pp. 681-688
- [Pannequin R., 2007] Pannequin R., *Proposition d'un environnement de modélisation et de test d'architectures de pilotage par le produit de systèmes de production*, Thèse de doctorat, Université Henri Poincaré, 2007
- [Pfeiffer A. et al., 2007] Pfeiffer A., Kadar B., Monostori L., *Stability-oriented evaluation of rescheduling strategies, by using simulation*, Computers in Industry, Volume 58, 2007, p. 630-643
- [Phillips L.-W. et al., 1976] Phillips L.-W., Unger P.-S., *Mathematical Programming Solution of a Hoist Scheduling Program*, IIE Transactions, Volume 8, Issue 2, 1976, pp. 219-225
- [Pinedo M., 2008] Pinedo M., *Scheduling: theory, algorithms, and systems*, 3ème édition, Springer, 2008

- [Poincaré H., 1892] Poincaré H., *Les méthodes nouvelles de la mécanique céleste*, Gauthiers-Villars, 1892
- [Pujo P. et al., 2008] Pujo P., Ounnar F., *Un modèle holonique et isoarchique pour le pilotage d'un job shop en flux tiré*, Conférence Internationale de MOdélisation et SIMulation, Paris, 2008
- [Rebollo M. et al., 2000] Rebollo M., Julian V., Carrascosa C., Botti V., *A Multi-Agent System for the Automation of a Port Container Terminal*, Rapport de recherche, Universidad Politecnica de Valencia, 2000
- [Rinnoy Kan A.-H.-G., 1976] Rinnoy Kan A.-H.-G., *Machine Scheduling Problems: Classification, Complexity and Computations*, Nijhoff, The Hague, 1976
- [Rossé-Bloch C., 1999] Rossé-Bloch C., *Contribution à l'ordonnancement dynamique des lignes de traitement de surface*, Thèse de doctorat, Université de Franche-Comté, 1999
- [Sabas A. et al., 2001] Sabas A., Delisle S., Badri M., *Vers une unification des méthodologies de développement des systèmes multiagents*, Journées francophones pour l'intelligence artificielle distribuée et les systèmes multi-agents, Montréal, 2001, pp. 327-330
- [Sabas A. et al., 2002] Sabas A., Badri M., Delisle S., *A Multidimensional Framework for the Evaluation of Multiagent System Methodologies*, 6th World Conference on Systemics, Cybernetics and Informatics, Orlando, Floride, USA, 2002, pp. I-211-216
- [Sammur C., 1996] Sammut C., *Automatic construction of reactive control systems using symbolic machine learning*, The Knowledge Engineering Review, Volume 11, 1996, pp. 27-42
- [Scheffel M., 2005] Scheffel M., *Optimal topology planning of optimcal networks with respect to overall design costs*, Optical Switching and Networking, Volume 2, 2005, pp. 239-248
- [Sénéchal O., 2004] Sénéchal O., *Pilotage des systèmes de production vers la performance globale*, Habilitation à Diriger des Recherches, Université de Valenciennes et du Hainaut Cambresis, 2004
- [Sgall J., 1998] Sgall J., *On-line Scheduling*, dans *Online Algorithms*, chap. 9, 1998, Berlin, Springer, pp. 196-231

- [Shapiro G.-W. et al., 1988] Shapiro G.-W., Nuttle H.-L.-W., *Hoist scheduling for a PCB electroplating facility*, IIE Transactions, Volume 20, Issue 2, 1988, pp. 157-167
- [Shilong W. et al., 2009] Shilong W., Wei Y., Jie Z., *Single Hoist Scheduling in no-wait flexible flow shop system*, International Symposium on Information, Huangshan, China, 2009, pp. 381-386
- [Smith R.-G., 1980] Smith R.-G., *The contract net protocol: High level communication and control in a distributed problem solver*, IEEE Transactions on Computer, Volume C29, Issue 12, 1980, pp. 1104-1111n3
- [Subaï C. et al., 2003] Subaï C., Niel E., Baptiste P., *Vers un pilotage propre des lignes de traitement de surface*, 4ème Conférence Francophone de MODélisation et SIMulation, Toulouse - France, 2003
- [Subaï C. et al., 2006] Subaï C., Niel E., Baptiste P., *Scheduling issues for environmentally responsible manufacturing: The case of hoist scheduling in an electroplating line*, International Journal of Production Economics, Volume 99, Issue 1-2, 2006, pp. 74-87
- [Sun J. et al., 2001] Sun J., Xue D., *A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources*, Computers in Industry, Volume 46, 2001, pp. 189-207
- [Trentesaux D., 2002] Trentesaux D., *Pilotage hétérarchique des systèmes de production*, Habilitation à Diriger des Recherches, Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines, 2002
- [Van Brussel H. et al., 1998] Van Brussel H., Wyns J., Valckenaers P., Bongaerts L., Peeters P., *Reference architecture for holonic manufacturing system: PROSA*, Computers in Industry, Volume 37, 1998, pp. 255-274
- [Van de Vonder S. et al., 2007] Van de Vonder S., Demeulemeester E., Herroelen W., *A classification of predictive-reactive project scheduling procedures*, Journal of Scheduling, Volume 10, 2007, pp. 195-207
- [Varnier C., 1996] Varnier C., *Extension du hoist scheduling problem cyclique. Résolution basée sur un traitement des contraintes disjonctives en programmation logique avec contraintes*, Thèse de doctorat, Université de Franche-Comté, 1996
- [Vin E. et al., 2010] Vin E., Delchambre A., Francq P., *A generalized cell formation problem solved by an adapted GGA*, MOSIM, Hammamet, Tunisie, 2010

- [Waldner J.-B., 1990] Waldner J.-B., *CIM, les nouvelles perspectives de la production*, Paris, Dunod-Bordas, 1990
- [Wasner M. et al., 2004] Wasner M., Zäpfel G., *An integrated multi-depot hub-location vehicle routing model for network planning of parcel service*, International Journal of Production Economics, Volume 90, 2004, pp. 403-419
- [White C.-H. et al., 1977] White C.-H., Wilson R. C., *Sequence dependent set-up times and job sequencing*, International Journal of Production Research, Volume 15, Issue 2, 1977, pp. 191-202
- [Yih Y., 1990] Yih Y., *Trace-driven knowledge acquisition (TDKA) for rule-based real time scheduling system*, Journal of Intelligent Manufacturing, Volume 1, 1990, pp. 217-230
- [Yih Y., 1994] Yih Y., *An algorithm for hoist scheduling problems*, International Journal of Production Research, Volume 32, Issue 3, 1994, pp. 501-516
- [Yin N. et al., 1992] Yin N., Yih Y., *Crane scheduling in a flexible electroplating line : a tolerance based approach*, Journal of Electronic Manufacturing, Volume 2, 1992, pp. 137-144
- [Zhao C. et al., 2010] Zhao C., Tang H., *Rescheduling problems with deteriorating jobs under disruptions*, Applied Mathematical Modelling, Volume 34, Issue 1, 2010, pp. 238-243
- [Zhu Y. et al., 2004] Zhu Y., Lim A., *Crane Scheduling with Spatial Constraints: Mathematical Model and Solving Approaches*, Naval Research Logistics, 2004
- [Zhu Y. et al., 2006] Zhu Y., Lim A., *Crane scheduling with non-crossing constraint*, Journal of the Operational Research Society, Volume 57, 2006, p. 1464–1471

**CONTRIBUTIONS A L'ORDONNANCEMENT REACTIF
DES INSTALLATIONS DE TRAITEMENT DE SURFACE
— APPLICATION INDUSTRIELLE —**

Résumé : Cette thèse de doctorat aborde le problème de l'ordonnancement réactif des installations de traitement de surface. L'émergence, sur le marché industriel, d'entreprises implantées dans des pays à faible coût de production et la réduction de l'horizon de planification de la production, imposent une révision du mode de pilotage. Depuis une décennie, des chercheurs travaillent sur le pilotage réactif, mode d'organisation dans lequel aucune décision n'est prise à l'avance. Ce mode de pilotage a l'avantage de répondre aux aléas de production et aux commandes volatiles, tout en assurant de bonnes performances. Il s'oppose principalement au pilotage prédictif dans lequel l'affectation des tâches aux différentes ressources se fait a priori.

Le traitement de surface est une étape de la production des pièces consistant à modifier la structure physico-chimique de la surface des pièces par immersion dans différents produits chimiques. Les pièces sont transportées par des ponts roulants, ressource généralement critique. Les temps de traitement, commençant avec l'immersion par le pont et se terminant par la reprise du pont, sont généralement bornés, ce qui impose au problème des contraintes temporelles fortes non présentes dans d'autres problèmes d'ordonnancement.

Cette thèse propose d'appliquer le mode de pilotage réactif à ces installations fortement contraintes. Après avoir démontré l'impossibilité d'une telle approche, parce qu'elle ne permet pas d'assurer la qualité des traitements appliqués aux produits, nous proposons une nouvelle solution au problème par l'utilisation couplée d'un ordonnancement réactif donnant la flexibilité recherchée et un ordonnancement prédictif assurant la qualité des produits. Cette thèse se termine par deux applications connexes à ce problème : la définition de la topologie d'une installation de traitement de surface à l'aide d'un algorithme génétique et l'aide au lancement des produits dans une telle installation.

La validation du concept scientifique développé a été réalisée dans un premier temps par simulation de flux. Le réalisme de la solution proposée a été montré par le déploiement de ces travaux au niveau industriel.

Mots clés : traitement de surface, couplage, pilotage réactif, pilotage prédictif, application industrielle

**REACTIVE HOIST SCHEDULING PROBLEM CONTRIBUTIONS
— INDUSTRIAL APPLICATION —**

Abstract : This thesis deals with the reactive hoist scheduling problem. The emergence of the industrial market companies located in countries with low production costs and reducing the planning horizon of production require a revision of the scheduling methods. Over the past 10 years, researchers are working on the reactive scheduling, method in which no decision is taken in advance. This method scheduling offers the advantage of responding to variations in production and variable requests, while ensuring a good performance. This is opposed to predictive scheduling, in which the allocation of tasks to different resources is done a priori.

The surface treatment is a step in the part production cycle dealing with modification of the physicochemical structure of the surface of parts by immersion in different chemicals. The parts are transported by crane (or hoist), which is generally the critical resource. The processing time, starting with dumping in the tanks and ending with the resumption of the hoist, are generally limited, which imposes strong temporal constraints, which are not present in other scheduling problems.

This thesis attempts to apply the reactive scheduling to these facilities constraints. Having demonstrated the impossibility of such an approach, we solve the problem by using a coupled reactive scheduling (assuming flexibility) and predictive scheduling (ensuring quality). This thesis concludes with two applications related to this problem: the definition of the topology of a surface treatment facility using a genetic algorithm and support for product order in such a facility.

The deployment at the industry level of this work is the best proof of the scientific concept developed.

Keywords: hoist scheduling problem, coupling, reactive, predictive, industrial application